

PROJECT ADMINISTRATION DATA SHEET

☒ ORIGINAL ☐ REVISION NO. _____Project No. E-21-621 (R5978-OAO)GTRC/XXDATE 7 / 12 / 85Project Director: Dr. William T. RhodesSchool/XX

EE

Sponsor: University of Dayton Research InstituteType Agreement: Purchase Order No. RI-39893 and Rev. No. 1 (Under Gov't Prime N00014-85-K-0479)Award Period: From 6/3/85 To 12/31/85 (Performance) 12/31/85 (Reports)

Sponsor Amount:

This Change

Total to Date

Estimated: \$ 99,849\$ 99,849Funded: \$ 99,849\$ 99,849Cost Sharing Amount: \$ NoneCost Sharing No: N/ATitle: Optical Computing Algorithms and Architectures

ADMINISTRATIVE DATA

OCA Contact

Brian J. Lindberg

X4820

1) Sponsor Technical Contact:

2) Sponsor Admin/Contractual Matters:

Dr. Eugene GerberSigmund W. BrzezickiUniversity of Dayton Research Inst.Purchasing Agent300 College Park Avenue, KL-465University of Dayton Research InstituteDayton, OH 45469300 College Park AvenueDayton, OH 45469Defense Priority Rating: DO-C9Military Security Classification: N/A(or) Company/Industrial Proprietary: N/A

RESTRICTIONS

See Attached N/A Supplemental Information Sheet for Additional Requirements.

Travel: Foreign travel must have prior approval - Contact OCA in each case. Domestic travel requires sponsor approval where total will exceed greater of \$500 or 125% of approved proposal budget category.

Equipment: Title vests with Gov't. except those items costing \$1,000 or less vest with GIT providing prior written approval to purchase received from Sponsor.

COMMENTS:



COPIES TO:

SPONSOR'S I. D. NO. 02.400.000.85.R08Project Director
Research Administrative Network
Research Property Management
AccountingProcurement/EES Supply Services
Research Security Services
Reports Coordinator (OCA)
Research Communications (2)GTRC
Library
Project File
Other Jones

SPONSORED PROJECT TERMINATION/CLOSEOUT SHEETDate 1/7/85Project No. E-21-621 (R5978-OA0)School EE

Includes Subproject No.(s) _____

Project Director(s) Dr. William T. RhodesGTRC / ~~XXX~~Sponsor University of Dayton Research InstituteTitle Optical Computing Algorithms and ArchitecturesEffective Completion Date: 12/31/85 (Performance) 12/31/85 (Reports)

Grant/Contract Closeout Actions Remaining:

- ☐ None
- ☒ Final Invoice or Final Fiscal Report
- ☒ Closing Documents
- ☒ Final Report of Inventions
- ☒ Govt. Property Inventory & Related Certificate
- ☐ Classified Material Certificate
- ☐ Other _____

Continues Project No. _____

Continued by Project No. _____

COPIES TO:

Project Director
Research Administrative Network
Research Property Management
Accounting
Procurement/GTRI Supply Services
Research Security Services
Reports Coordinator (OCA)
Legal Services

Library
GTRC
Research Communications (2)
Project File
Other M. Heyser

A. JonesR. Embry

5-21-60
Georgia Institute of Technology
School of Electrical Engineering
Atlanta, Georgia 30332

William T. Rhodes, Project Director and Principal Investigator
Thomas K. Gaylord, Principal Investigator

Research on Optical Computing Algorithms and Architectures

Abstract

Research on new architectural and algorithmic approaches to optical computing has been conducted in the areas of 1) optical degrees of freedom and devices for controlling them, 2) ultra-short optical pulses and nonlinear optics, 3) number representations, 4) content-addressable-memory processors, and 5) integrated optical Givens rotation devices. Results and recommendations are presented.

Technical Summary

1. Objectives

The principal objective of this research program is the conceptual development of new architectural and algorithmic approaches to ultra-highspeed computing using optical and opto-electronic techniques.

2. Description of Work Performed and Results

2.1 Optical Degrees of Freedom and Devices for Controlling Them

A review was conducted of the properties of light that can be controlled in an optical computer (e.g., polarization, propagation direction, wavelength, amplitude, phase, intensity), means for controlling them, and the advantages and disadvantages of different methods, including speed of operation and energy consumption. This was done to provide the basis for a study of optical computer architectures unprejudiced by notions of what basic light control operations should be employed and to provide to as great an extent as possible for flexibility in conceptual architectural design.

2.2 Ultra-Short Optical Pulses and Nonlinear Optics [1]

A preliminary study has been conducted of ways in which nonlinear optical phenomena can be used with ultrashort optical pulses to enhance the capabilities of optical computers. Ultrashort pulses are of interest because of their potential for exploiting the full available bandwidth of the optical source. It has been determined that nonlinear optical interactions can be used in various ways to allow for the cascade of highspeed content-addressable-memory-based optical computing systems and to compensate for loss and aberrations. Nonlinear optical interactions that exhibit both high speed and high efficiency (both of which are necessary for optical computer systems) are achievable only with optical pulses of high peak power. Mode-locked lasers and optical pulse compression systems produce such pulses every nanosecond or so. Unfortunately, schemes proposed thus far for exploiting the full temporal frequency bandwidth of the light source [2] result in reduced peak optical power and, hence, reduced efficiency in the nonlinear interactions. Attention is now being given to methods for avoiding this problem.

2.3 Number Representation [3]

Preliminary results have been obtained in the investigation of number representations for optical computing systems. Binary coding, multilevel coding, and residue number systems have been analyzed in terms of the primitive operations of addition and multiplication. Examples of fixed-radix and residue number representations have been calculated with and without multilevel coding. A detailed comparison has been made for the case of 16-bit full precision addition and multiplication. This example has indicated a clear advantage of

using multilevel coding.

2.4 Content-Addressable Memory Processors [4]

Preliminary results have been obtained showing the use of optical content-addressable memory processors in non-primitive operations such as discrete matched filtering (cross correlation). The design of an optical holographic truth-table look-up system that processes multilevel coded numbers has been developed.

2.5 Integrated Optical Givens Rotation Device [5]

The Givens rotation operation plays a central role in matrix formulations of linear algebraic signal processing. A design concept for an integrated optical device that implements this operation has been developed. The device uses electronically-controlled thick grating diffraction to control optical wave amplitudes in accord with the desired rotation operation. It has been shown that existing electro-optic phase shifting and grating diffraction devices can be combined to produce an extremely fast Givens rotation device. Operations that can be performed by such a device include matrix triangularization, matrix inversion, solution of least squares problems, singular value decomposition, and the calculation of eigenvalues and eigenvectors.

3. Conclusions and Recommendations

It is premature at this stage to draw many conclusions. However, in terms of program direction we think that the content-addressable-memory work is particularly important, for it shows promise for optical computer architectures capable of exploiting both the spatial and the temporal potential of optics. Particular attention should be given to multiple-input-multiple-output systems because of their significance in parallel processing generally.

The Givens rotation device is important of itself because of its possible applications. However, it is also significant because of the way it exploits natural physical phenomena for performing operations not easily performed on a binary-logic-based electronic computer. The basic approach discussed in an attachment needs to be studied further in terms of accuracy and speed achievable, and related architectures and algorithms should be investigated.

Nonlinear optics used in conjunction with ultrashort optical pulses can in principle solve many of the problems associated with wideband operation of cascable logic-based optical computer subsystems. There is, however, a conflict between the need for high peak-power optical pulses (for high-efficiency nonlinear interactions) and the temporal modulation of the light waves necessary for exploiting the full optical bandwidth. This conflict must be studied further and somehow resolved. Further, the energy and efficiency characteristics of nonlinear optical devices under development should be considered in connection with specific (e.g., strawmen) systems.

References

1. W. T. Rhodes and J. A. Buck, "Optical Computing and Nonlinear Optics," to be presented at and appear in the proceedings of the SPIE conference 625 on Optical Computing, Los Angeles, January 23-24 1986. (In preparation.)
2. H. J. Caulfield and Tomas Hirshfeld, "Optical Communication at the Source Bandwidth Limit," **Applied Optics**, vol. 16, pp. 1184-1186, May 1977.
3. T. K. Gaylord and M. M. Mirsalehi, "Truth-Table Look-Up Processing: Number Representation, Multi-Level Coding, and Logical Minimization," **Optical Engineering**, vol. 25, pp. xxx-xxx, January 1986 (accepted). (Copy attached.)
4. M. M. Mirsalehi and T. K. Gaylord, "Multi-Level Coded Residue-Based Content-Addressable-Memory Optical Computing," submitted to **Applied Optics**. (Copy attached.)
5. M. M. Mirsalehi, T. K. Gaylord, and E. I. Verriest, "Integrated Optical Givens Rotation Device," submitted to **Applied Optics**. (Copy attached.)

Technical Discussion

Copies of references 3, 4, and 5 are attached.

Author: PLEASE read the proofs against your typewritten manuscript. You are responsible for verifying the accuracy of spelling, math, numerical data, the figures and legends, and references.

Issue date: January 1986
Return proofs, ms., figures, & forms by
Oct 23, 1985

Galley 1

OT-103/Gaylord

Opt. Eng. January 1986 mg

Disk 1 103-1

Truth-table look-up processing: number representation, multilevel coding, and logical minimization

T. K. Gaylord, MEMBER SPIE
M. M. Mirsalehi, MEMBER SPIE
Georgia Institute of Technology
School of Electrical Engineering
Atlanta, Georgia 30332

Abstract. The need for ultra-high-speed computing for a variety of modern processing problems has generated new interest in using truth-table look-up techniques. Further, due to the frequently parallel nature of these processing problems, optical systems appear to be promising for these applications. The basic principles of truth-table look-up processing are reviewed in this paper. The issues of number representation, multilevel coding, and logical minimization are discussed. Example fixed-radix and residue number representations are given with and without multilevel coding. Logical reduction techniques are discussed with examples. A comparison of the number of truth-table entries needed for 16-bit full-precision addition and multiplication is given, illustrating the advantage of the multilevel coded residue number representation.

Subject terms: digital optical computing; truth-table look-up processing; optical data processing; number representation.

Optical Engineering 25(1), 000-000 (January 1986).

Has this paper been
published at any conference
published in any other
journal? Editor

Invited Paper OT-103 received July 24, 1985; revised manuscript received Aug. 28, 1985; accepted for publication Aug. 30, 1985; received by Managing Editor Sept. 23, 1985.
© 1986 Society of Photo-Optical Instrumentation Engineers.

CONTENTS

1. Introduction
 - 1.1. The need for ultra-high-speed computing
 - 1.2. Truth-table look-up processing as a possible solution
 - 1.3. Issues associated with truth-table look-up processing
2. Truth-table look-up processing architectures
 - 2.1. Location-addressable memories
 - 2.2. Content-addressable memories
 - 2.3. Hardware logic gates
3. Number representation
 - 3.1. Fixed-radix system
 - 3.2. Residue number system
4. Multilevel coding
 - 4.1. Encoding
 - 4.2. Example systems
5. Logical minimization
 - 5.1. Forms of logical reduction results
 - 5.2. Finding prime implicants
 - 5.3. Constructing table of choice
 - 5.4. Obtaining minimal sum
6. Optical implementation
7. Discussion and summary
8. Acknowledgments
9. References

1. INTRODUCTION

1.1. The need for ultra-high-speed computing

The number of areas in need of computing power well beyond that currently available is large and increasing. High throughput computing systems (or, equivalently, ultra-high-speed systems) are needed in areas such as adaptive antenna beam forming, artificial intelligence, remote sensing, ultra-high-resolution image processing, control of communication networks, aeronautical design, seismic data interpretation, meteorology, air-traffic control, synthetic-aperture radar imaging, missile guidance, defense early-warning systems, and molecular, nuclear, and plasma physics simulations.¹⁻⁴ For example, real-time computation of images from synthetic-aperture radar data would require the equivalent of several trillion multiplications per second.^{5,6}

1.2. Truth-table look-up processing as a possible solution

Many functions, transformations, and operations may be represented by a binary truth table in which the outputs are given for all possible input combinations. Direct implementation of processors from a truth-table representation has not been common in the history of data and signal processing. This is largely due to the numerous efficient algorithms that can be programmed on general-purpose Von Neumann type computers. However, the types of problems listed above are largely beyond the capabilities of present-day computing systems. These problem areas have emphasized the growing need for parallel application of the same algorithm to large arrays of data. This, in turn, has generated renewed interest in the direct implementation of truth-table-based processors.

Many of these processing problems are highly complex and computationally intensive. However, the solutions can frequently be expressed in terms of matrix-based algorithms.⁷ In these, a single operation is repeated many times over many elements. This highly regular nature lends itself naturally to parallel processing and to truth-table look-up techniques. The pronounced structure of the algorithms has not been efficiently utilized in past data processing systems.

There are three general architectures for truth-table implementation. These involve using (1) location-addressable memory, (2) content-addressable memory, and (3) hardware logic gates. These basic architectures are discussed in Sec. 2. Gate arrays and programmable array logic (PAL) devices that are in widespread use today are electronic implementations of truth tables. In another example, off-line a priori calculations are used to prestore in memory the controllers for given speed ranges for a fighter aircraft. This is necessary since the required calculations cannot be performed in real time and thus are obtained by look-up. Papachristou⁸ has presented an encoding scheme for a direct truth-table implementation of discrete and residue-based functions (see Sec. 3.2) that employs PAL devices. Truth-table look-up has been used for changing the function in optical cellular logic to implement two-dimensional logical neighborhood functions for applications such as digital image processing.⁹ Look-up methods have been used to find the correct mappings required to implement a residue matrix-vector multiplier.¹⁰ Discrete matched filtering and other functions can be implemented by truth-table look-up techniques.¹¹ Ishihara¹² has described the use of truth-table look-up in the design of optical processing systems by the joint university-governmental-industrial Optical Computer Group of the Japanese Society of Applied Physics. Potentially, many complex problems can be treated with look-up methods.

med /

i /

1.3. Issues associated with truth-table look-up processing

The viability of truth-table look-up processing for a particular application depends on a number of critical issues. These include (1) architectural implementation, (2) number representation, (3) number encoding, and (4) logical reduction and/or minimization. In an electronic or optical hardware logic gate implementation, the resulting number of gates and the number of interconnections are determined by the truth-table representation used. Perhaps more important, the needed routes of the interconnections are prescribed by the final form of the truth table used. In a bulk optical configuration using a content-addressable memory, the truth table used specifies the amount of storage needed (e.g., number of holographically recorded reference patterns).¹³ For this case, however, the number and form of the interconnections specified are of no particular significance since the interconnections are made optically in three-dimensional space and their routing is automatically taken into account in the original design of the system. This is in dramatic contrast to very-large-scale integration (VLSI) in integrated circuits, in which the form of the interconnections is typically the limiting factor in the design of complex systems.

2. TRUTH-TABLE LOOK-UP PROCESSING ARCHITECTURE

2.1. Location-addressable memories

The most straightforward implementation of a truth table may be achieved by the storage of the entire truth table in a direct, or location-addressable, memory (LAM) such as an electronic read-only memory (ROM). These systems require a memory size (in bits) of

$$S = 2^p q, \quad (1)$$

where p is the number of input bits and q is the number of output bits. In these processors, the inputs determine the address of the answer.

2.2. Content-addressable memories

Less storage is generally required when a truth table is implemented using a content-addressable memory (CAM). Such memories may utilize electronic, magnetic, optical, or other technologies. The unity-result truth tables for each output bit are stored in the CAM. A unity-result or a null-result truth table may be constructed from those combinations of inputs that cause a particular output bit to be a "one" or a "zero," respectively. The unity-result truth table represents the canonical sum-of-products expression for the logical function corresponding to each output bit.

OE Jan 1986

In a content-addressable memory, inputs are compared with the stored tables, and detected matches cause the appropriate output bits to be a "one" (if a unity-result truth table is stored) or a "zero" (if a null-result truth table is stored). The stored input words (or "reference patterns" in pattern recognition terminology) are the function minterms in the sum-of-products expression (unity-result truth table) or the function maxterms in the product-of-sums expression (null-result truth table). The number of function minterms for each output bit for addition and multiplication using the residue number system has been compiled.¹⁴ This number is always less than or equal to the number of function maxterms due to the inherent structure associated with the operations of addition and multiplication. In the optical holographic implementation of content-addressable memory, the number of function minterms represents the number of holograms that need to be stored in the system.¹⁰ Using thick holographic recording media, such as photorefractive lithium niobate, holograms may be multiplexed together in a common volume¹⁵ with the number of possible stored holograms being on the order of a thousand.¹⁶

2.3. Hardware logic gates

A truth table may also be implemented through the direct use of Boolean logic gates. Each binary output variable, when represented as a sum of products (or product of sums) of binary input variables, may be implemented with three levels of logic in the form of a programmable array logic device. For a sum-of-products form, the sequence of logic gates is NOT, AND, OR. For a product-of-sums form, it is NOT, OR, AND. The number of function minterms represents the number of AND gates (in sum-of-products implementation) that must be formed to realize each output bit.

3. NUMBER REPRESENTATION

3.1. Fixed-radix system

In many ways, the manner in which numbers are represented places subtle and fundamental limitations on what types of calculations can be efficiently performed on them. One need only try to do calculations with Roman numerals to realize the impact and importance of the positional (Arabic) number system. Innovations such as the concept of zero as a position-holding digit, negative numbers, and the representation of fractions in a positional number system required centuries of evolutionary development.¹⁷ It is only a sense of provincialism that causes the decimal system to be viewed as a uniquely appropriate number system. Historically, satisfactory progress in many areas of mathematics and engineering has been limited by number representation.¹⁸ Number systems and number representation are again becoming the subject of increasing study for achieving faster and more efficient computation.

In a fixed-radix number system, any number N_b in a radix (base) b may be represented by a_n, \dots, a_{-m} , such that

$$N_b = \sum_{i=-m}^n a_i b^i, \quad (2)$$

where a_i is any of the b digits allowed. The integers n and m control the size and precision of the number. The base for the binary, octal, decimal, and hexadecimal number systems are 2, 8, 10, and 16, respectively. The ranges of digits are 0 and 1, 0 through 7, 0 through 9, and 0 through 9 and A through F, respectively. For example, $1101_2 = 15_8 = 13_{10} = D_{16}$.

A property of all fixed-radix number systems is the interdependence of digit results in numerical operations. In addition and multiplication this is manifested as a carry digit propagating from lower to higher significant digits. This requires that the most significant bit of a result cannot be known until calculation of all lesser significant bits has been completed. Thus, carry propagation represents a fundamental limitation for high-speed digital electronic data processing systems.

In truth-table look-up processing, all digits of the answer can be calculated simultaneously. However, carry propagation produces a very undesirable effect in these processors. Because the output digits depend on all lesser significant input digits, the truth tables can become enormous. As the number of input digits increases, the size of the resulting truth table increases exponentially. Clearly, a number system without interdigit dependence would be highly desirable to avoid these unmanageably large truth tables. The residue number system described below has no such interdigit dependence.

3.2. Residue number system

Unlike the commonly used decimal and binary number systems, the residue number system (RNS) is an unweighted system. The base of a residue system is chosen as n relatively prime (containing no common factors) numbers m_1, m_2, \dots, m_n , called moduli. Any integer X is then represented as an n -tuple (x_1, x_2, \dots, x_n) , where $x_i = X \wedge_{m_i}$ (meaning $X \bmod m_i$). This representation is unique if the range of X is less than or equal to M , where

$$M = \prod_{i=1}^n m_i, \quad (3)$$

and represents the dynamic range. Negative numbers can be included by an arbitrary partitioning of the range of the number system.

Size 7 /

Size 7 /

5 /

2 /

1 (int. Lnc) // 21 t /

Size 7 /

Size 7 /

OE Jan '86

The important feature of RNSs is that the fixed-point arithmetic operations can be performed on each digit individually. That is, if $X = (x_1, x_2, \dots, x_n)$ and $Y = (y_1, y_2, \dots, y_n)$ are two numbers of the same system, then $Z = X * Y = (z_1, z_2, \dots, z_n)$, where $z_i = (x_i * y_i)_{m_i}$ for $i = 1, 2, \dots, n$, and $*$ represents the addition, subtraction, or multiplication operation. Division may be performed but it is difficult,^{19,20} except for the remainder zero case.

As an illustrative example, consider a set of four moduli $\{3, 4, 5, 7\}$. In this system, the decimal numbers $X = 23$ and $Y = 14$ are represented as $X = (2, 3, 3, 2)$ and $Y = (2, 2, 4, 0)$. The results of performing addition, subtraction, and multiplication on these numbers are $X + Y = (1, 1, 2, 2)$, $X - Y = (0, 1, 4, 2)$, and $X \times Y = (1, 2, 2, 0)$, which are the residue representations of the correct answers, i.e., 37, 9, and 322, respectively.

In residue arithmetic, there are a number of basic operations that are difficult to perform. These are division, scaling, sign detection, overflow detection, and relative-magnitude determination. In spite of these difficulties, the fact that the calculations associated with different moduli are independent of each other makes RNSs suitable for parallel processing. An especially significant increase in the number of operations per second is achieved when the calculations are composed of residue addition and multiplication (e.g., in matrix-matrix multiplication or discrete Fourier transformation).

The cyclic nature of residue arithmetic makes it particularly suitable for optical implementations. Using the cyclic property of a phase of the polarization of light, a number of numerical optical residue processors have been developed.²¹⁻²⁶ Hughes Research Laboratories has constructed an electronic residue arithmetic digital image understanding system (RADIUS) that performs 5×5 pixel generalized convolution operations on 8-bit pixels.²⁷ The moduli used are 31, 29, 23, and 19, the four largest primes less than $2^5 = 32$. Additions and multiplications are performed at high speed by truth-table look-up of the residues for each radix from a random-access memory (RAM). Binary-to-residue and residue-to-binary conversion are also accomplished by truth-table look-up.

Moduli selection is an important issue in the design of any system based on residue arithmetic. Many system parameters are affected by the moduli set, and conflicting requirements may make the selection difficult. For example, to reduce the execution time for all operations that involve a mixed-radix conversion, it is desirable to have as few moduli as possible; hence, large moduli are preferred. On the other hand, the hardware of most systems increases rapidly with the size of the moduli; therefore, using a large number of small moduli has the advantage of decreasing the complexity of the system.

There are other considerations that imply different selections,¹⁹ such as increasing the storage efficiency, having unity multiplicative inverses, and having unity multipliers in the Chinese remainder theorem. A procedure for selecting the moduli that are optimum in the sense of requiring the minimum number of reference patterns is given in Ref. 14.

y/
1/2 it bar // -2 fls /

PE 3

AA —

As mentioned previously, the lack of interdigit dependence makes RNSs potentially extremely useful in reducing the required number of reference patterns that need to be stored. This is a very powerful feature in content-addressable memory applications. For example, consider the addition of two 16-bit numbers. If the usual binary system is used to represent the numbers, a total of 36,507,222,016 reference patterns, each a 32-bit word, are needed to be stored in a CAM for truth-table look-up processing. However, using the moduli set {4,5,7,9,11,13}, the number decreases to only 694 patterns of 4-bit to 8-bit words. As shown in subsequent sections, further reduction in the number of reference patterns can be obtained by applying multilevel coding and logical minimization techniques.

4. MULTILEVEL CODING

4.1. Encoding

Multilevel coding has recently been used as a technique for further reducing the number of truth-table entries (reference patterns) that need to be stored.¹¹ Multilevel coding is an extension of binary coding in which more than two levels are used. For example, in three-level (ternary) coding, the integers zero to eight are represented as 00, 01, 02, 10, 11, 12, 20, 21, and 22, respectively. Minimization of multilevel coded reference patterns requires a type of logic different from the commonly used binary logic. The appropriate logic, known as multiple-valued logic, is an active area of research today.

Although significant progress has been made in the theoretical aspects of multiple-valued logic, there have been only a small number of electronic implementations of this logic. The first full-scale three-value electronic computer was completed in 1958 at Moscow State University in the Soviet Union.²⁸ Electronic ternary logic has been used in constructing arithmetic units.²⁹ Multiple-valued operation of integrated circuits has been investigated at the Naval Research Laboratory.³⁰ Currently, for example, the Intel 8087 floating-point processor uses four levels of current in ROMs. In optics, shadow-casting techniques have been used to implement multiple-valued logic.³¹ The fact that there have not been more implementations is partly due to the difficulties in realizing multilevel devices and partly due to the significant progress that has been achieved in the area of binary logic systems. However, as has recently been shown¹¹ multilevel coding in some optical systems can be implemented as easily as binary coding.

4.2. Example systems

Some examples of decimal, binary, residue, binary-coded residue, multilevel coded residue, and binary-coded multilevel-coded residue number representations are presented in Table I.

5. LOGICAL MINIMIZATION

5.1. Forms of logical reduction results

Procedures for the logical reduction of a truth table may produce results in a variety of forms. For logical functions expressed as a sum of products, these include (1) the near-minimal sum, (2) the minimal sum, and (3) the absolute minimal sum. A *near-minimal sum* is generally obtained by using a nonexhaustive reduction technique. In these methods, the sum-of-products logical expression is greatly reduced but not necessarily minimized. These techniques can be very fast computationally because they do not consider all reduction possibilities.³² A *minimal sum* is a reduced sum-of-products expression that has the minimum possible number of terms in it. These forms are often called "sloppy minimal sums" in the literature.³³ An *absolute minimal sum* is a reduced sum-of-products expression that has both the minimum possible number of terms in it and the minimum number of factors (or variables) in each term. This form is also called the "real minimal sum" and sometimes (confusingly) the "minimal sum."

As an example, the logical minimization for the simple case of addition modulus 4 is shown in Fig. 1. The method illustrated in this figure represents only one particular approach to logical minimization. However, in general, the steps in minimization are (1) define the initial truth table for the operation in question; (2) find all prime implicants; (3) construct the table of choice; and (4) obtain a minimal sum. Each of these steps will be discussed in subsequent sections. An alternative technique, the use of the Karnaugh map,³⁴ allows a minimal sum to be obtained directly without using the steps listed here. It is a graphical method that allows the minimal sum to be visualized directly, but it is impractical for functions of more than about five variables. The steps listed above, however, can be programmed on a computer and can handle any number of input variables.

5.2. Finding prime implicants

As shown in Fig. 1, the function is first specified and then coded. In this case, since the modulus is 4, binary coding is used directly. From these results, the truth tables for the most significant bit (MSB) and least significant bit (LSB) may be constructed directly. Then the logical expressions may be written. All of the prime implicants³⁵ of a logical function can be determined by the Quine-McCluskey method.^{35,36} This method is summarized in numerous textbooks³⁷ and is illustrated in Fig. 1 for the MSB. The minterms are listed in subgroups starting with those that have a single "one" in them, then those with two "ones" in them, and so on, until all minterms are listed in the first group. Then, all pairs of minterms that differ by only one factor are checked. These pairs are listed in a second group with "don't care" dashes at the location of the differing factor for the pair. The process of combining terms that differ by only one factor is then continued until no further combining is possible. For the example in Fig. 1, no further combining is possible in the second group. All unchecked terms in the prime implicant table constitute the list of all prime implicants.

21
21

33

As the number of variables increases, the Quine-McCluskey method of determining the prime implicants becomes inefficient in terms of execution time and required memory. More efficient methods include the Tison algorithm³³ and the tree-structured approach of Morreale.³⁸ An even more efficient modified tree structure method has been developed recently by Guest.³⁹

5.3. Constructing table of choice

In the Quine-McCluskey method, after the prime implicants have been determined, a table of choice is constructed. This consists of all the prime implicants (listed vertically in Fig. 1) and all the function minterms (listed horizontally in Fig. 1). Each prime implicant row is marked in the columns of the minterms covered by that prime implicant. Thus, it is observed how the entries in the initial truth table are covered by the prime implicants.

5.4. Obtaining minimal sum

A minimal sum is obtained by finding the minimal prime implicant covering of the table of choice. This is referred to in the literature as the covering problem, the set-covering problem, or the minimum-covering problem.³³ An absolute minimal sum may be found using the following steps. Note that this absolute minimal sum may not be unique; there may be other absolute minimal sums that can be obtained by changing the order in which the selections below are made.

Step one: Select essential rows. Some rows uniquely cover some of the columns. These rows must be selected in order to cover those columns. The prime implicants associated with these rows are called essential prime implicants. The essential rows and all columns that contain marks in these rows should be eliminated from the table of choice.

Step two: Eliminate dominated rows. One row dominates a second row if the first row has marks in all columns in which the second row has marks. If the dominating row has the same or fewer variables in its prime implicant, the prime implicant associated with the dominated row should be eliminated from the table of choice. If a minimal sum, rather than an absolute minimal sum, is satisfactory, then the number of variables in the prime implicants need not be compared.

Step three: Eliminate dominating columns. Similarly, one column dominates a second column if the first column has marks in all rows in which the second column has marks. The minterms associated with the dominating columns should be eliminated from the table of choice.

Step four: Repeat steps one through three until all columns are eliminated. The resulting sum of all essential row prime implicants represents the absolute minimal sum.

There are cases, however, in which the above procedure is unable to eliminate all columns. The remaining table, which contains at least two marks in each column, is called a cyclic table. In this case, the tabular method using a recursive branch-and-bound algorithm presented by Muroga³³ may be used to obtain the absolute minimal sum.

se verify this
ing.

Minimization techniques in multiple-valued logic are somewhat different from those used in binary logic. In binary logic, if two terms in a sum-of-products expression are the same in all bit positions except one, they can be combined into one term that has a "don't-care" bit at that location. For example, 100 and 101 can be combined as $10X$, where X represents a "don't-care" bit. In multiple-valued logic, terms can be combined in several ways. For example, in ternary logic, the terms 120, 121, and 122 can be reduced to $12X$, where X (referred to as a "complete-don't-care" digit) represents a digit with possible values of 0, 1, and 2. If one of the above terms is absent, the other two can still be combined. For example, the terms 120 and 121 can be reduced to $12X_{01}$, where X_{01} (referred to as a "partial-don't-care" digit) represents a digit with possible values of 0 and 1, but not 2.

As the number of entries in a truth table increases, the minimization procedure becomes too complex to be handled by hand. Associated with the present work, a computer program has been developed to reduce the reference patterns for an arbitrary level coding and to obtain the minimum number of required patterns. The Quine-McCluskey technique was extended to handle the multiple-valued logic case. In the first part of the program, a complete list of the prime implicants is obtained. Using this set, a table of choice is constructed. Then, a minimal sum set is obtained by applying the reduction rules to the table. The results for residue addition and multiplication for moduli 2 through 32 are given in Ref. 11. These results show that the number of reference patterns can be decreased significantly if the appropriate level of coding is used. If the modulus can be expressed as $M = p^n$, where p is a prime number and n is a positive integer greater than one, p -level coding is the best choice. For example, binary coding is appropriate for moduli such as 4, 8, 16, and 32, while ternary coding is beneficial for moduli such as 9 and 27. This is due to the highly regular structures of the truth tables that are produced in these cases. For a modulus that is not expressible in the above form, the proper coding level can be found among its prime factors. The prime factor that produces the largest contribution to the modulus is usually the best choice. For example, binary coding is appropriate for modulus $12 (= 2^2 \times 3)$, while modulus $6 (= 2 \times 3)$ benefits from ternary coding.

The optimum sets of moduli for minimizing the number of reference patterns for performing 16-bit full-precision addition and multiplication are given in Table II. Results for both binary-coded residue numbers and multilevel coded residue numbers are given before and after logical minimization.

6. OPTICAL IMPLEMENTATION

An optical implementation of a truth-table look-up data processing system using a holographic content-addressable memory is described in Ref. 11. The optical system presented is capable of processing multilevel-coded numbers. The operations of addition, multiplication, and discrete matched filtering (cross-correlation) are evaluated in terms of the number of required reference patterns for various word lengths.

① / snail // ② / snail //

7. DISCUSSION AND SUMMARY

Truth-table look-up processing concepts, implementations, and applications have been reviewed. Due to current and future needs for ultra-high-speed computing, there has been an increasing interest in using truth-table look-up techniques. Further, due to the parallel nature of numerous modern processing problems, optical systems are serious candidates for these applications.

The issues of number representation, multilevel coding, and logical minimization have been discussed. The number of entries (reference patterns) in the reduced truth table is of central importance in determining the viability of look-up techniques for a particular application. In hardware logic gate implementations (electronic or optical), the number of gates and the number of interconnections are prescribed by the logically reduced form of the truth table used. In the location-addressable memory and content-addressable memory implementations, the reduced truth table specifies the amount of storage required. In these cases, the interconnections are of no particular significance, in marked contrast to the hardware logic gate case. In an optical content-addressable memory, the truth-table look-up processing can be performed in parallel.

For comparison, the number of truth-table entries for 16-bit full-precision addition and multiplication are given in Table III. Results are supplied for binary and residue representations. Values are given with and without logical minimization and with and without multilevel coding. The dramatic reduction using residue number systems is apparent. Further significant reductions are shown by using logical minimization and multilevel coding. Thus, number representation, multilevel coding, and logical minimization are all significant factors in truth-table look-up processing.

8. ACKNOWLEDGMENTS

This work was supported in part by a grant from the Strategic Defense Initiative Office, administered through the Office of Naval Research, and by a grant from the Joint Services Electronics Program.

Opt. Eng. now includes titles of journal and proceedings papers and book chapters in the references. If readily available, on a separate page please provide titles for each reference.

Galley 13

OT-103/Gaylord

Opt. Eng. January 1986 mg

Disk 1 103-13

Authors needed for Refs. 2-47 Are page numbers needed for Refs. 2 & 47

9. REFERENCES

1. A. L. Robinson, *Science* 203(4376), 156 (1979).
2. *Computer* 16(6), (1983).
3. *Phys. Today*, special issue on Advances in Computers for Physics, 37(5), 61 (1984).
4. *Proc. IEEE*, special issue on Supercomputers—Their Impact on Science and Technology, 72(1), (1984).
5. W. M. Brown, *IEEE Trans. Aerospace Electron. Sys.* AES-3(2), 217 (1967).
6. W. M. Brown, G. G. Houser, and R. E. Jenkins, *IEEE Trans. Aerospace Electron. Sys.* AES-9(2), 166 (1973).
7. K. Bromley, "An interview with Keith Bromley on signal processing," *Optical Engineering Reports* No. 14, p. 1, SPIE (February 1985).
8. C. A. Papachristou, *IEEE Trans. Comput.* C-32(10), 961 (1983).
9. T. Yatagai, S. Inaba, H. Nakano, and M. Suzuki, "Automatic flatness tester for very large scale integrated circuit wafers," *Opt. Eng.* 23(4), 401 (1984).
10. S. F. Habiby and S. A. Collins, *Topical Meeting on Optical Computing*, Technical Digest, pp. TuD41-TuD44, OSA, Washington (1985).
11. M. M. Mirsalehi and T. K. Gaylord, submitted to *Appl. Opt.*
12. S. Ishihara, *Topical Meeting on Optical Computing*, Technical Digest, p. TuE21, OSA, Washington (1985).
13. C. C. Guest and T. K. Gaylord, *Appl. Opt.* 19(7), 1201 (1980).
14. C. C. Guest, M. M. Mirsalehi, and T. K. Gaylord, *IEEE Trans. Comput.* C-33(10), 927 (1984).
15. J. E. Weaver and T. K. Gaylord, "Evaluation experiments on holographic storage of binary data in electro-optic crystals," *Opt. Eng.* 20(3), 404 (1981).
16. D. L. Staebler, W. J. Burke, W. Phillips, and J. J. Amodei, *Appl. Phys. Lett.* 26(4), 182 (1975).
17. J. R. Newman, *The World of Mathematics*, pp. 430-520, Simon & Schuster, New York (1956).
18. R. M. Kline, *Digital Computer Design*, p. 19, Prentice-Hall, Englewood Cliffs, NJ (1977).
19. N. S. Szabo and R. I. Tanaka, *Residue Arithmetic and Its Applications to Computer Technology*, McGraw-Hill, New York (1967).
20. E. Kinoshita, H. Kosako, and Y. Kojima, *IEEE Trans. Comput.* C-22(2), 134 (1973).
21. A. Huang, in *Proc. Int. Optical Computing Conference*, pp. 14-18, IEEE, New York (1975).
22. A. Huang, Y. Tsunoda, J. W. Goodman, and S. Ishihara, *Appl. Opt.* 18(2), 149 (1979).
23. A. Tai, I. Cindrich, J. R. Fienup, and C. C. Aleksoff, *Appl. Opt.* 18(16), 2812 (1979).
24. S. A. Collins, Jr., "Numerical optical data processor," in *Effective Utilization of Optics in Radar Systems*, B. W. Vatz, ed., *Proc. SPIE* 128, 313 (1977).
25. J. N. Polky, D. D. Miller, and R. L. Gutmann, "Optical residue arithmetic data processing," Boeing Aerospace Company, Report AFWAL/AADO-2, Wright Patterson AFB, Ohio (1982).
26. J. Jackson and D. Casasent, *Appl. Opt.* 22(18), 2817 (1983).
27. S. D. Fouse, G. R. Nudd, and A. D. Cummings, in *Proc. 6th Int. Conf. on Pattern Recognition*, pp. 262-269, IEEE, New York (1982).
28. D. C. Rine, ed., *Computer Science and Multiple-Valued Logic*, North-Holland, Amsterdam (1977).
29. P. Sebastian and Z. G. Vranesic, in *Proc. 1972 Symp. on Theory and Applications of Multiple-Valued Logic*, SUNY, Buffalo, New York (1972).
30. G. Abraham, *Computer* 7(9), 42 (1974).
31. R. Arrathoon and S. Kozaitis, "Shadow-casting for multiple-valued associative logic," *Opt. Eng.* 25(1), (1986).
32. Z. Arevalo and J. G. Bredeson, *IEEE Trans. Comput.* C-27(11), 1028 (1978).
33. S. Muroga, *Logic Design and Switching Theory*, p. 163, John Wiley & Sons, New York (1979).
34. M. Karnaugh, *IEEE Trans. Commun. & Electron.* 72(5), 593 (1953).
35. W. V. Quine, *Am. Math. Monthly* 59(10), 521 (1952).
36. E. M. McCluskey, *Bell Sys. Tech. J.* 35(6), 1417 (1956).
37. T. L. Booth, *Digital Networks and Computer Systems*, John Wiley & Sons, New York (1971).
38. E. Morreale, *IEEE Trans. Electron. Comput.* EC-16(5), 611 (1967).
39. C. C. Guest, "Holographic optical digital parallel processing," Ph.D. thesis, Georgia Institute of Technology (1983).

run down /

AIEE

FIGURE 1

Fig. 1. The process of logical minimization for residue addition modulus 4 is illustrated, including the steps of defining the initial truth table, finding all prime implicants, constructing the table of choice, and finding the absolute minimal sum.

0 5 2 A A —

Galley 14

OT-103/Gaylord

Opt. Eng. January 1986 mg

Disk 1 103-14

TABLE I. Number Representation Examples

Number representation	Decimal	Binary	Residue	Binary-coded residue	Multilevel coded residue	Binary-coded multilevel coded residue
Digit weight	10 1	64 32 16 8 4 2 1	-	-	-	-
Moduli	-	-	9 5 4	9 5 4	9 5 4	9 5 4
Coding level	-	-	-	-	3 5 2	3 5 2
Digit maximum value	99	1111111	8 4 3	1000 100 11	22 4 11	10 10 100 1 1
A	7	0000111	7 2 3	0111 010 11	21 2 11	10 01 010 1 1
B	14	0001110	5 4 2	0101 100 10	12 4 10	01 10 100 1 0
A + B	21	0010101	3 1 1	0011 001 01	10 1 01	01 00 001 0 1
A × B	98	1100010	8 3 2	1000 011 10	22 3 10	10 10 011 1 0

Paste-up:
Run back, delete
lead, and
extend underline

TABLE II. Optimum Sets of Moduli, Coding Level, and Numbers of Reference Patterns for Performing 16-Bit Full-Precision Addition and Multiplication Using Binary-Coded and Multilevel-Coded (Allowing for Two-, Three-, and Five-Level Coding) Residue Number Systems*

	Addition			Multiplication		
	Modulus	Coding level	Number of reference patterns	Modulus	Coding level	Number of reference patterns
Binary-coded residue (without logical reduction)	4	2	16	5	2	20
	5	2	25	7	2	54
	7	2	63	9	2	102
	9	2	117	11	2	170
	11	2	187	13	2	264
	13	2	286	16	2	392
Binary-coded residue (after logical minimization)				17	2	528
				19	2	666
				23	2	1056
	3	2	6	5	2	15
	5	2	19	7	2	18
	7	2	36	9	2	55
Multilevel-coded residue (without logical reduction)	11	2	90	11	2	84
	13	2	116	13	2	115
	16	2	60	16	2	144
				17	2	205
				19	2	266
				23	2	381
Multilevel-coded residue (after logical minimization)	4	3 or 5	12	5	5	16
	5	5	20	7	5	42
	7	5	49	9	3 or 5	84
	9	5	99	11	5	140
	11	5	154	13	5	216
	13	5	234	16	5	328
Multilevel-coded residue (after logical minimization)				17	5	400
				19	5	522
				23	5	792
	4	2	8	5	2 or 3	15
	5	3	18	7	2	18
	7	2	36	9	3	30
Multilevel-coded residue (after logical minimization)	9	3	36	11	5	78
	11	3	89	13	3	105
	13	5	113	16	2	44
				17	3	175
				19	5	242
				23	3	360

*Addition of two 16-bit words produces a 16-bit sum with an output carry bit (no input carry bit). Multiplication of two 16-bit words produces a full 32-bit product (rather than the simpler fixed-point or floating-point representations).

FL /

underline /

scratch line ///

4 /

Galley 16

OT-103/Gaylord

CE Jan 86 11g

Disk 1 103-16R

TABLE III. Comparison of Number of Required Reference Patterns to Perform 16-Bit Full-Precision Addition and Multiplication Using Various Encoding Schemes*

	Addition	Multiplication
Binary (without logical reduction)	3.65×10^{10}	6.32×10^{10}
Binary (after logical minimization)	3.28×10^5	1.43×10^7
Binary-coded residue (without logical reduction)	694	3252
Binary-coded residue (after logical minimization)	327	1183
Multilevel-coded residue (without logical reduction)	568	2540
Multilevel-coded residue (after logical minimization)	300	1067

*Two-, three-, and five-level coding have been used in the multilevel coding results.

PE — AA — 1

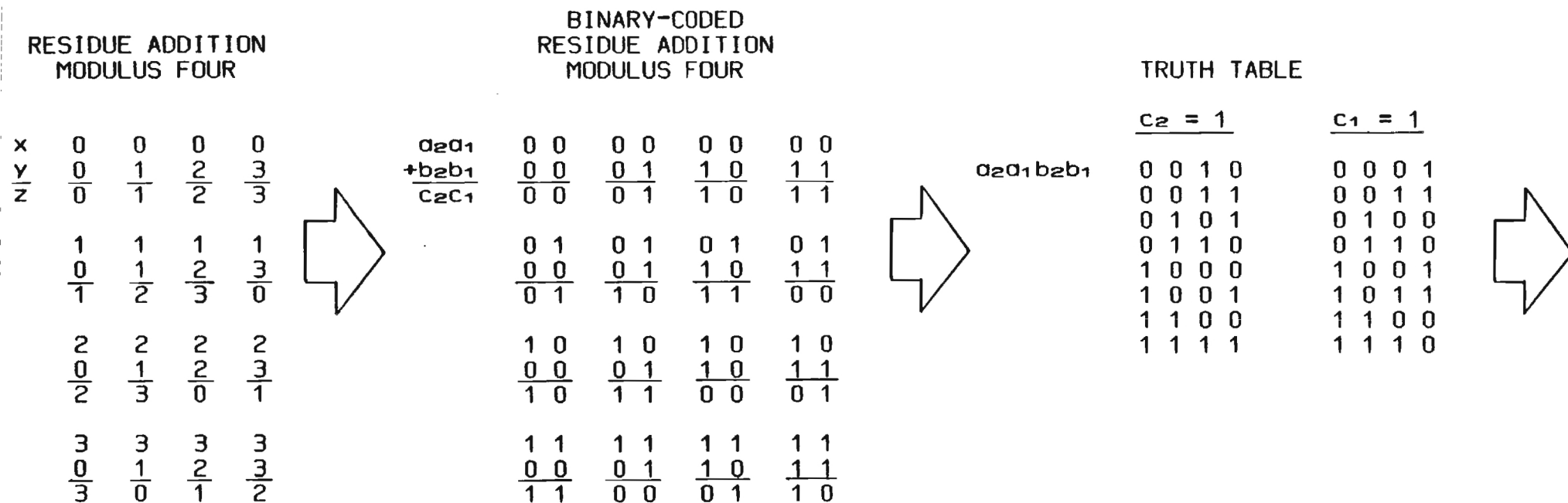


Fig. 1 (top portion)

LOGICAL EXPRESSIONS

PRIME IMPLICANT TABLE

$$\underline{c_2 = 1}$$

PRIME IMPLICANTS

$$\underline{c_2 = 1}$$

$$\begin{aligned} &\bar{a}_2\bar{a}_1b_2\bar{b}_1 + \bar{a}_2\bar{a}_1b_2b_1 + \bar{a}_2\bar{a}_1\bar{b}_2b_1 + \\ &\bar{a}_2a_1b_2\bar{b}_1 + \bar{a}_2a_1b_2b_1 + \bar{a}_2\bar{a}_1\bar{b}_2b_1 + \\ &a_2a_1\bar{b}_2\bar{b}_1 + a_2a_1b_2b_1 = c_2 \end{aligned}$$



FIRST GROUP

SECOND GROUP

$$a_2a_1b_2b_1$$

$$a_2a_1b_2b_1$$

$$\begin{array}{cccc} 0 & 0 & 1 & 0 \checkmark \\ 1 & 0 & 0 & 0 \checkmark \end{array}$$

$$\begin{array}{cccc} 0 & 0 & 1 & - \\ 0 & - & 1 & 0 \\ 1 & 0 & 0 & - \end{array}$$

$$\begin{array}{cccc} 0 & 0 & 1 & 1 \checkmark \\ 0 & 1 & 0 & 1 \end{array}$$

$$\begin{array}{cccc} 1 & - & 0 & 0 \end{array}$$

$$\begin{array}{cccc} 0 & 1 & 1 & 0 \checkmark \\ 1 & 0 & 0 & 1 \checkmark \end{array}$$

$$\begin{array}{cccc} 1 & 1 & 0 & 0 \checkmark \end{array}$$

$$\begin{array}{cccc} 1 & 1 & 1 & 1 \end{array}$$



$$a_2a_1b_2b_1$$



$$\begin{array}{cccc} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & - \\ 0 & - & 1 & 0 \\ 1 & 0 & 0 & - \\ 1 & - & 0 & 0 \end{array}$$

Fig. 1 (upper middle portion)

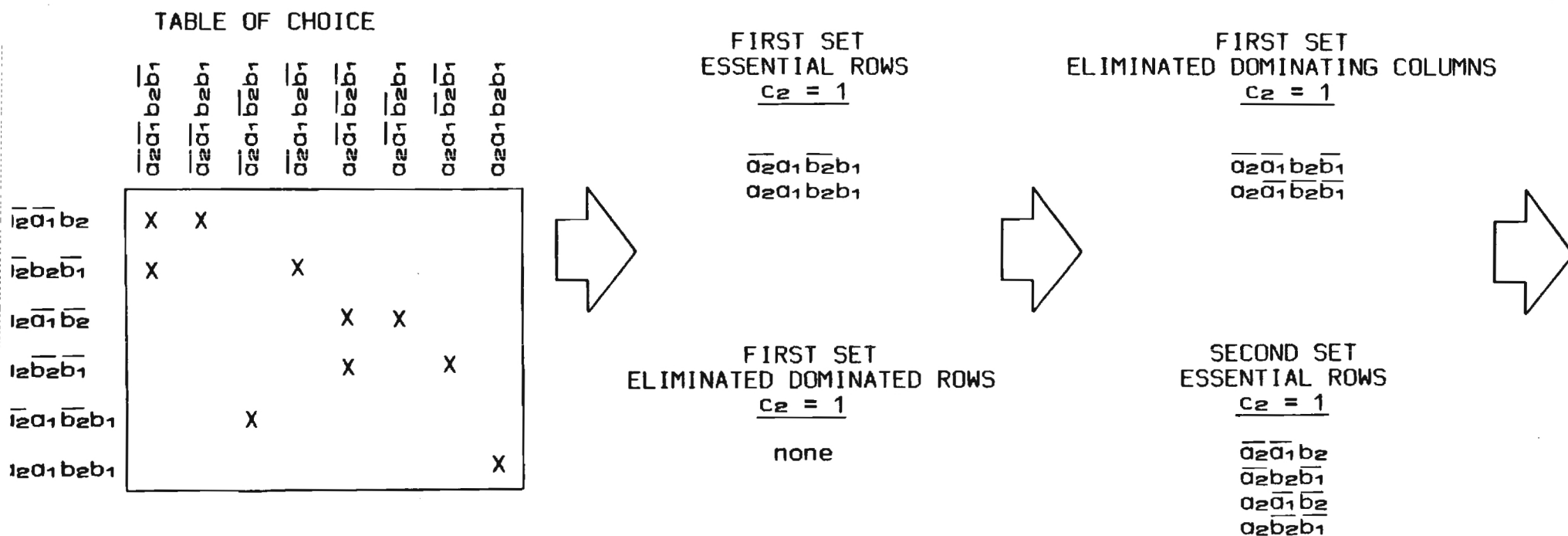


Fig. 1 (lower middle portion)

REDUCED LOGICAL EXPRESSIONS

$$\overline{a_2}\overline{a_1}b_2 + \overline{a_2}b_2\overline{b_1} + a_2\overline{b_2}\overline{b_1} +$$

$$a_2\overline{a_1}\overline{b_2} + \overline{a_2}a_1\overline{b_2}b_1 + a_2a_1b_2b_1 = c_2$$

$$a_1\overline{b_1} + \overline{a_1}b_1 = c_1$$



$$a_2a_1b_2b_1$$

REDUCED TRUTH TABLE

<u>c₂ = 1</u>				<u>c₁ = 1</u>			
0	0	1	X	X	1	X	0
0	X	1	0	X	0	X	1
1	X	0	0				
1	0	0	X				
0	1	0	1				
1	1	1	1				

Fig. 1 (bottom portion)

Truth-Table Look-Up Parallel Data Processing
Using an Optical Content-Addressable Memory

M. M. Mirsalehi and T. K. Gaylord
School of Electrical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

Truth-table look-up data processing using a holographically implemented content-addressable memory is described. The implementations of the addition, multiplication, and discrete matched filtering (cross-correlation) are presented. It is shown that multi-level coding can be used to reduce the number of reference patterns in content-addressable memories. The number of reference patterns required to implement residue addition and multiplication operations are provided for moduli 2 through 32 and for 2-, 3-, and 5-level coding. An optical truth-table look-up processor based on multi-level coding scheme is presented.

I. Introduction

A. The Need for Optical Digital Computing

In spite of recent advances in electronic computers, there exists a number of problem areas¹ such as meteorology, aerodynamics, molecular dynamics, fusion energy, and finite-element analysis, that demand computing powers well beyond what is currently available. The major difficulty (known as the von Neumann bottleneck²) in increasing the computing power of existing electronic computers, is the sequential nature of the processing. To overcome this difficulty, parallel processing architectures are under development for the next generation of computers. Optical processors, in contrast to electronic systems, are inherently parallel; they can perform operations on the elements of a two-dimensional array, simultaneously. Optical digital systems combine the parallelism and speed of optics with the accuracy and flexibility of a digital system. As such, they are promising candidates for increasing computing power.

B. Associative and Content-Addressable Memories

In precise computer engineering terminology³, an associative memory and a content-addressable memory have different capabilities. An associative memory allows recall with incomplete or imperfect input information. It also takes into account input contextual information that may be present and provides a single output answer. A content-addressable memory, on the other hand, implements a type of associative recall. However, the recall is based on a complete parallel search of all reference patterns stored in the memory. Thus, the output can be a parallel array of answers, as opposed to a single output answer in the case of an associative memory. Thus, in general,

a content-addressable memory is considered to be a single-input/parallel-output system.

With optics, a content-addressable memory can be implemented in a straightforward manner by holographically recording many reference patterns using the large dynamic range associated with volume holographic recording media such as electro-optic crystals⁴. In addition, however, the basic single-input/parallel-output character of a content-addressable memory can be expanded. The lack of angular selectivity in the direction perpendicular to the recording plane of incidence of volume holograms⁵ allows an entire array of parallel inputs to be processed simultaneously by the same set of holographic gratings. An optical volume holographic content-addressable memory can produce a separate set of parallel outputs for each of the inputs. Therefore, such a memory is capable of performing as parallel-input/parallel-output system. This, in turn, produces the possibility of very high throughput parallel computing.

C. Truth-Table Look-Up Processing

Digital operations can be implemented by 1) converting the operation into a sequence of logic steps that can be realized by logic gates, or 2) reading the output from a memory in which the truth table corresponding to that operation has been previously stored. In spite of the superiority of the latter method in speed and flexibility, it has not been common in the history of data processing. This is mainly due to the large sizes of the memories that are usually needed. Recent technological advances and a growing need for parallel processing have generated renewed interest in the direct implementation of truth-table look-up processing.

There are two methods of implementing a truth table. One method

is to store each output bit in a memory location whose address is determined by the input word. This type of memory is called a location-addressable memory (LAM). In the second method, for each output bit, all the input bit combinations that produce a "one" in that output bit location are stored. During the processing step, the input bits are compared with all the stored reference patterns that correspond to each output bit. If a match is detected, that particular output bit is considered to be a "one", otherwise, it is a "zero". This type of memory is called a content-addressable memory (CAM). Content-addressable memories can benefit from logical reduction techniques, hence they usually require much less storage than LAM's.

As mentioned above, one of the advantages of the truth-table look-up technique is its flexibility. Any discrete function or operation whose truth table is known can be implemented by this technique. This includes addition, multiplication, division, exponentiation, series evaluation, and other operations. For example, truth-table look-up processing may be used to implement discrete matched filtering. Consider a one-dimensional discrete reference signal, $g(n)$, of four units in length, defined as: $g(n) = n$ for $1 \leq n \leq 4$. Assume that it is desired to detect the signals similar to $g(n)$. First, all possible input signals are normalized so that the resulting discrete signals, $f(n)$, have energies that are the closest possible values to (but not greater than) the energy of the reference signal, $g(n)$. Second, the cross-correlations between the normalized signals, $f(n)$, and the reference signal are obtained (Fig. 1). If the two signals are the same, then the cross-correlation operation becomes an autocorrelation and a maximum peak value is produced. If the signals are different, the result will have a lower peak value of its cross-correlation. The height of the peak depends on the

similarity of the two signals. For the above example, the autocorrelation has a peak of 30. It might be of interest to detect not only the exact matched pattern, but also the patterns that have a high cross-correlation (for further inspection). In this case, in addition to $g(n)$, these patterns are also stored in the content-addressable memory. For example, if patterns of four units in length that produce a cross-correlation peak equal to or greater than 28 are desired, then 12 reference patterns need to be stored. The output of the system can be represented by a two-bit number. An exact match could be indicated by making the first bit equal to a "one", and a mismatch with a high cross-correlation peak (28 or 29 in the above example) could be indicated by making the second bit equal to a "one". In the case of a mismatch with low cross-correlation peak, both output bits would be "zero". Thus all possible input signals can be checked in a cross-correlation sense using only a few stored reference signals (12 in the above example). This is representative of truth-table look-up processing: a large number of calculations are done in advance in order to construct a relatively small truth-table that can then be used repeatedly to perform a calculation on all possible future inputs.

II. Truth-Table Representation

A. Truth-Table Reduction

As mentioned in the previous section, the major problem that faces the truth-table look-up processing technique is the large number of reference patterns that usually need to be stored. This has prevented the implementation of electronic look-up processors except for simple cases. Although, holographic memory systems with large storage capacities exist, the amount of data storage required by

CAM's is still typically larger. For example, to implement the full-precision addition of two 16-bit numbers with a CAM based on the usual binary number system, a total of 36,507,189,248 reference patterns are required to be stored. This is dramatically beyond the number of holograms that can be recorded in state-of-the-art holography. Two methods of reducing the number of reference patterns that have been previously investigated are: 1) Using residue number system (RNS)⁴, and 2) applying logical minimization techniques⁶.

B. Residue Number System

A residue number system is defined by choosing n relatively prime numbers, m_1, m_2, \dots, m_n , called moduli. Any integer X , can then be represented as an n -tuple, (x_1, x_2, \dots, x_n) , where $x_i = |X|_{m_i}$ (read X modulo m_i) is the least positive integer remainder that is obtained from the division of X by m_i . For example, consider a four-modulus system with moduli 3, 4, 5, and 7. In this system, the decimal numbers $X = 23$ and $Y = 14$ are represented as $X = (2, 3, 3, 2)$ and $Y = (2, 2, 4, 0)$, respectively. The important feature of RNS is that fixed-point arithmetic operations can be performed on each digit individually. For example, the results of performing addition and multiplication on the above numbers are $X + Y = (1, 1, 2, 2)$ and $X \cdot Y = (1, 2, 2, 0)$. These are the residue representations of the correct answers, i.e., 37 and 322, respectively. For more information on residue arithmetic, the reader is referred to Ref. 7. The fact that the digits of a residue number are independent of each other results in a number of small truth tables rather than a single large truth table. Consequently, the number of reference patterns for a particular function is significantly reduced using RNS. For example, choosing the moduli set $M = \{4, 5, 7, 9, 11, 13\}$, and using binary coding to represent the digits,

the number of reference patterns corresponding to full-precision addition of two 16-bit words is reduced from 36,507,189,248 to only 694.

C. Logical Minimization

Logical minimization techniques, such as the Karnaugh map method, the Quine-McCluskey table method, or the partitioned list method, can be used to obtain further reduction in the number of reference patterns. Minimization results for residue addition and multiplication using binary coding representation have been reported in Ref. 6.

D. Multi-Level Coding

Multi-level coding has been recently used as another technique for further reducing the number of reference patterns⁸. Here, this technique is described in detail and new results are presented.

Multi-level coding is an extension of binary coding in which more than two levels are used. For example, if three-level (ternary) coding is used, the integers zero to eight will be represented as 00, 01, 02, 10, 11, 12, 20, 21, and 22, respectively. Minimization of multi-level coded reference patterns requires a type of logic different from the commonly used binary logic. The appropriate logic, known as multiple-valued logic, is an active area of research today. Although, significant progress has been made in the theoretical aspects of multiple-valued logic⁹, electronic implementations of this logic have only recently begun to appear. This is partly due to the difficulties in realizing multi-state devices, and partly due to the significant progress that has been achieved in the area of binary logic systems. However, as it will be shown in the next section,

multi-level coding in optical systems can be implemented as easily as binary coding.

Minimization techniques in multiple-valued logic are somewhat different from those used in binary logic. In binary logic, if two terms in a sum-of-products logical expression are the same in all bit positions except one, they can be combined into one term which has a "don't-care" bit at that location. For example, 100 and 101 can be combined as 10X, where X represents a don't-care bit. In multiple-valued logic, terms can be combined in several ways. For example, in ternary logic, the terms 120, 121, and 122 can be reduced to 12X, where X (herein referred to as a "complete-don't-care" digit) represents a digit that can take any possible value (in this case 0, 1, and 2). If one of the above terms is absent, the other two can still be combined. For example, the terms 120 and 121 can be reduced to 12X₀₁, where X₀₁ (herein referred to as a "partial-don't-care" digit) represents a digit with possible values of 0 and 1, but not 2.

As the number of entries in a truth table increases, the minimization procedure becomes too complex to be handled by hand. In the present work, a computer program has been developed to reduce the reference patterns for an arbitrary level coding and to obtain the minimum number of required patterns. The Quine-McCluskey technique¹⁰ was extended to handle the multiple-valued logic case. In the first part of the program, a complete list of the prime implicants is obtained. Using this set, a table of choice is constructed. Then, a minimal sum set is obtained by applying the reduction rules to the table. The results for residue addition and multiplication for moduli 2 through 32 are given in Table 1. These results show that the number of reference patterns can be decreased significantly if the appropriate level of coding is used. If the modulus can be expressed

as $M = p^n$, where p is a prime number and n is a positive integer greater than one, p -level coding is the best choice. For example, binary coding is appropriate for moduli such as 4, 8, 16, and 32, while ternary coding is beneficial for moduli such as 9 and 27. This is due to the highly regular structures of the truth tables that are produced in these cases. For a modulus that is not expressible in the above form, the proper coding level can be found among its prime factors. The prime factor that produces the largest contribution to the modulus is usually the best choice. For example, binary coding is appropriate for modulus 12 ($= 2^2 \times 3$), while modulus 6 ($= 2 \times 3$) benefits from ternary coding.

III. Optical Implementation

A. NAND-Based Processing

The optical implementation described here is a modified version of the NAND-based processor that has been previously introduced by Guest and Gaylord⁴. The main advantage of this processor is its capability of operating as a parallel-input/parallel-output system. For more information on the binary-coded version of the processor the reader is referred to Ref. 4.

A schematic diagram of a ternary-coded NAND processor is shown in Fig. 2. The input array is composed of three parts: two input words and one reference bit. Each digit of the input words has two corresponding positions in the input array. If the digit has a value of "1", it is coded as a transparent aperture in the first position. Similarly, a "2" is coded as a transparent aperture in the second position. A "0" is coded as opaque apertures in both positions. In general, if n -level coding is used, each digit can have any integer value between zero and $n-1$ and it can be positional coded with $n-1$

spatial locations.

B. Recording

To implement a particular operation, all the input word combinations that produce a nonzero value at each output digit must be stored. These reference patterns are recorded as thick holograms in a photorefractive crystal, such as LiNbO_3 . The recording process for each reference pattern is accomplished in three steps. Figure 3 shows the procedure for recording a reference pattern (2110). First, all the positions of the input array that are complementary to the reference pattern are made transparent and a hologram is recorded. The relative phase between the reference beam and the object beam in this recording is considered the reference phase for the other steps and it is assigned a value of zero. Then, the phase of one of the beams is shifted by 180° and a hologram of all positions that correspond to the reference pattern is recorded. Finally, the reference bit is recorded at a relative phase of zero. The exposure period for the first two recordings are the same, resulting in an amplitude diffraction efficiency of η_a for each recorded bit. The reference bit, however, is recorded with an amplitude diffraction efficiency of $R\eta_a$, where R is the number of nonzero digits in the reference pattern (3 for the above case).

The phasor diagram corresponding to the above example is shown in Fig. 4a. All the three steps of recording are performed with the reference beam incident at a particular angle. For other reference patterns, the position of the reference beam is changed so that each pattern is recorded with a different angle between the beams. When all the reference patterns for a particular function or operation are recorded, the recording process is complete and the processor is

ready to implement that function or operation.

C. Playback (Data Processing)

During the reading process, the positions of the input array that correspond to the two input words and the reference bit are made transparent, while the other locations are made opaque. The light passing through the transparent apertures upon diffraction by the holograms reconstructs the reference beams at different angles. Depending on the phase of the recorded bits, these diffracted beams are added to or subtracted from each other at the detector elements. For each output digit, if the input pattern matches one of the corresponding reference patterns, the diffracted beams cancel each other at the element of the detector array that corresponds to the matched pattern. As a result, a dark spot is produced at that element. This is detected electronically and the proper value is assigned to the corresponding output digit. If the input pattern is not similar to one of the reference patterns, the diffracted beams do not completely cancel out each other at any of the elements of the detector array that correspond to that output digit. This represents a value of "0" for that digit. Possible output values for the example studied above are shown in Fig. 4b. The numbers indicate the degeneracy of each case.

D. "Don't-Care" Digits

The recording process for different types of don't-care digits are presented in Fig. 5. In the case of a complete-don't-care digit, the locations that correspond to that digit are made opaque during all the three steps of recording (Fig. 5a). As a result, during the reading process, the presence or absence of light at those locations has no effect on the reconstructed wavefront. Due to the positional

coding scheme that has been used to represent each digit, two types of partial-don't-care digits should be distinguished. These are: 1) partial-don't-care digits that include zero as an allowed value, and 2) partial-don't-care digits that do not include zero as an allowed value. To record a partial-don't-care digit of the first type, the locations that correspond to disallowed values of that digit are recorded at 0° phase, while the locations that correspond to the allowed values are made opaque during all the recording steps (Fig. 5b). In the case of a second type partial-don't-care digit, the locations that correspond to the disallowed values are recorded at 0° phase and those that correspond to the allowed values are recorded at 180° (Fig. 5c).

IV. Summary and Discussion

Optical truth-table look-up parallel data processing has been reviewed. By recording the appropriate input patterns, any function that can be described by a truth table may be implemented. This produces the advantage that all the calculations are done in advance and the results are used to construct the minimized truth table (set of reference patterns). The optical system operates as a content-addressable memory and makes a parallel comparison of the input words with the prestored reference patterns. Using the lack of angular selectivity in the direction perpendicular to the recording plane of incidence allows a parallel array of input words to be processed. Thus the processor performs as a parallel-input/parallel-output system.

The operations of addition, multiplication, and discrete matched filtering (cross-correlation) were described. The effect of coding level on the number of required reference patterns in a residue

based content-addressable memory was studied. It was found that for moduli expressible as $M = p^n$, where p is a prime number and n is a positive integer greater than one, p -level coding is the most efficient scheme. This is due to the significant reduction in the number of reference patterns that can be obtained by applying logical minimization techniques. In general, the prime factors that divide a modulus can be used to find the coding level that corresponds to the minimum number of reference patterns. The reduction techniques for multi-level coded reference patterns were described and the number of reference patterns required for residue addition and multiplication operations was provided for moduli 2 through 32, and for 2-, 3-, and 5-level coding.

The results presented in Table I were used to study some practical cases of addition and multiplication operations. For each case, the moduli set was selected such that it covered the required range with minimum number of reference patterns. The results are presented in Table II. The number of reference patterns in this table is equal to the number of holograms and also to the number of elements of the output photo-detector array in the optical implementation described. These numbers show that the operations studied can be implemented with state-of-the-art holography.

REFERENCES

1. H. J. Caulfield, S. Horvitz, G. P. Tricoles, and W. A. Von Winkle, Eds., Proc. IEEE, Special Issue on Optical Computing, 72 (1984).
2. R. Bernhard, "Computers: Emphasis on Software," IEEE Spectrum 17, 32 (1980).
3. T. Kohonen, Associative Memory - A System-Theoretical Approach (Springer-Verlag, Berlin, 1977).
4. C. C. Guest and T. K. Gaylord, "Truth-Table Look-Up Optical Processing Utilizing Binary and Residue Arithmetic," Appl. Opt. 19, 1201 (1980).
5. H. J. Gallagher, T. K. Gaylord, M. G. Moharam, and C. C. Guest, "Reconstruction of Binary-Data-Page Holograms for an Arbitrary Oriented Reference Beam," Appl. Opt. 20, 300 (1981).
6. C. C. Guest, M. M. Mirsalehi, and T. K. Gaylord, "Residue Number System Truth-Table Look-Up Processing - Moduli Selection and Logical Minimization," IEEE Trans. Comput. C-33, 927 (1984).
7. N. S. Szabo and R. I. Tanaka, Residue Arithmetic and Its Applications to Computer Technology (McGraw-Hill, New York, 1967).
8. M. M. Mirsalehi and T. K. Gaylord, "Multi-Level Coded Residue-Based Content-Addressable Memory Optical Computing," in Topical Meeting on Optical Computing, Technical Digest, Sponsored by Optical Society of America, March 18-20, 1985, Incline Village, Nevada, p. WB1-1.

9. D. C. Rine, Ed., Computer Science and Multiple-Valued Logic (North-Holland, Amsterdam, 1977).
10. e. g., S. Muroga, Logic Design and Switching Theory (John Wiley & Sons, New York, 1979), p. 163.

FIGURE CAPTIONS

Fig. 1. An example of discrete matched filtering; (a) input signal, $f(n)$, different from the reference signal, $g(n)$; (b) input signal same as the reference signal. The symbol $*$ represents cross-correlation.

Fig. 2. Schematic diagram of a ternary-coded NAND processor that implements residue multiplication modulo 6. D: detector; E: electrooptic crystal; L: Fourier transform lens; LSD: least significant digit; MSD: most significant digit; OB: object beam; RB: reference beam.

Fig. 3. Recording procedure for a ternary-coded reference pattern (2110). (a) Recording the complementary pattern ($122\bar{X}_n$) at 0° relative phase. (b) Recording the reference pattern (2110) at 180° relative phase. (c) Recording the reference bit at 0° relative phase.

Fig. 4. (a) Example phasor diagram corresponding to a recorded reference pattern (2110) in a ternary-coded NAND processor. The location of each recorded phasor at the input array is indicated with two numbers (m,n). The first number is the digit number (m=1 for the least significant digit). The second number specifies a particular position of that digit. The vertical separations between the phasors are artificially made in order to distinguish the phasors from each other. (b) Phasor diagram showing possible wavefront amplitudes at the detector element corresponding to the above reference pattern. Numbers indicate the degeneracies of the phasors.

Fig. 5. Recording procedures for patterns with don't-care digits. (a) A pattern with a complete-don't-care digit (00X1). (b) A pattern with a partial-don't-care digit of the first type (011X₀₁). (c) A pattern with a partial-don't-care digit of the second type (1X₁₂). The first and third recordings are performed at 0° relative phase, while the second is performed at 180° relative phase.

Table 1. Number of Required Reference Patterns for Residue Addition and Multiplication Using Different Levels of Coding.

Modulus	Addition			Multiplication		
	2-level	3-level	5-level	2-level	3-level	5-level
2	2	2	2	1	1	1
3	6	6	6	4	4	4
4	8	12	12	5	8	8
5	19	18	20	15	15	16
6	26	18	30	19	11	17
7	36	37	39	18	28	29
8	24	46	49	14	32	38
9	64	36	60	55	30	50
10	64	72	38	58	65	28
11	90	89	93	84	85	78
12	80	84	103	71	74	85
13	116	124	113	115	105	116
14	118	138	130	101	138	119
15	124	110	74	136	118	68
16	60	158	173	44	178	163
17	180	158	207	205	175	297
18	172	74	221	209	74	221
19	224	205	242	266	256	242
20	176	223	128	200	277	125
21	272	201	270	342	244	299
22	260	291	280	308	356	342
23	288	309	293	381	360	381
24	204	232	318	238	279	390
25	343	307	200	494	441	282
26	329	291	385	426	453	485
27	377	150	400	557	180	524
28	311	377	426	393	562	591
29	400	403	481	621	640	652
30	371	358	350	510	513	371
31	350	504	532	780	762	735
32	136	533	539	143	806	782

Table II. Appropriate Moduli Sets and Number of Required Reference Patterns (N_r) for Addition and Multiplication Operations.

Operation	Moduli Set	N_r
4-bit full-precision addition	3, 4, 5	32
8-bit full-precision addition	3, 5, 7, 8	84
12-bit full-precision addition	3, 5, 7, 8, 11	173
16-bit full-precision addition	4, 5, 7, 9, 11, 13	300
16-bit fixed-point addition	3, 5, 7, 8, 11, 13	288
32-bit fixed-point addition	5, 7, 9, 11, 13, 16, 17, 19, 23	1001
4-bit full-precision multiplication	3, 4, 5, 7	42
8-bit full-precision multiplication	5, 7, 9, 13, 16	212
12-bit full-precision multiplication	5, 7, 9, 11, 13, 17, 32	584
16-bit full-precision multiplication	5, 7, 9, 11, 13, 16, 17, 19, 23	1067
16-bit fixed-point multiplication	3, 5, 7, 8, 11, 13	234
32-bit fixed-point multiplication	5, 7, 9, 11, 13, 16, 17, 19, 23	1067

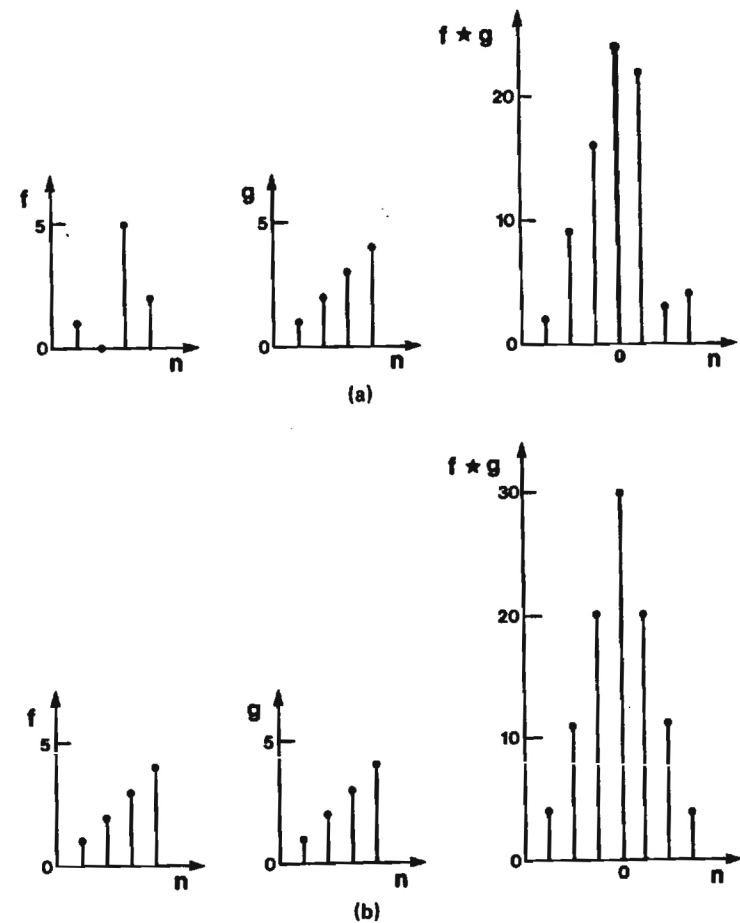


Fig.1

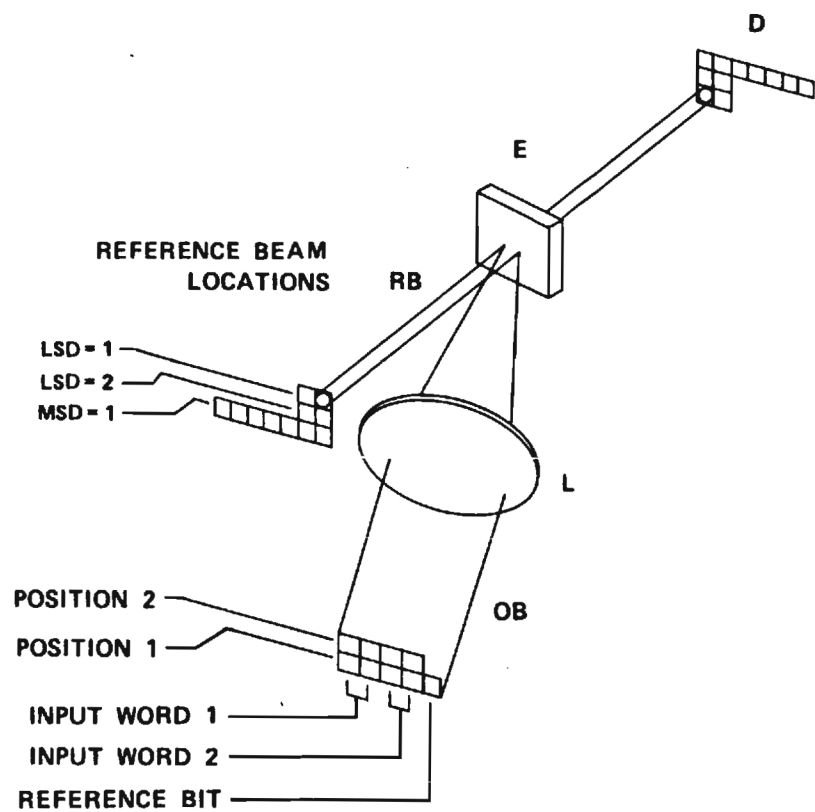


Fig. 2

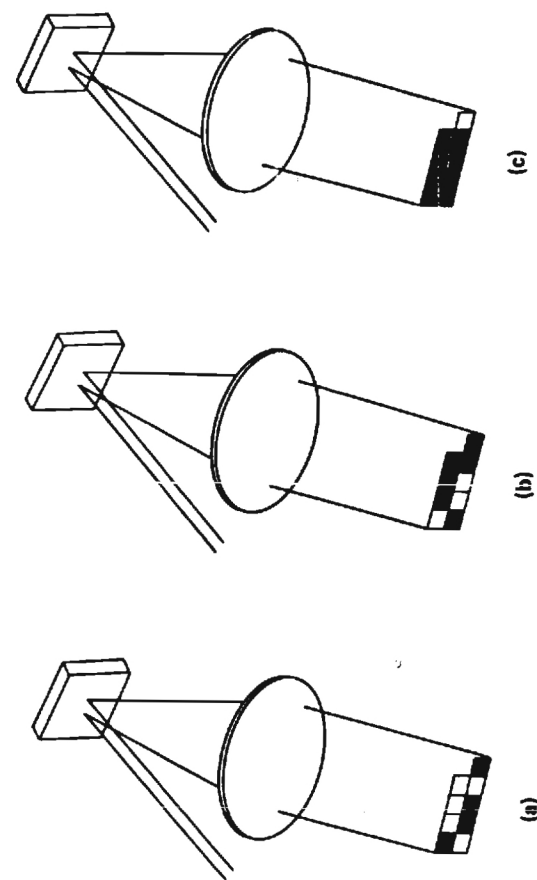


Fig. 3

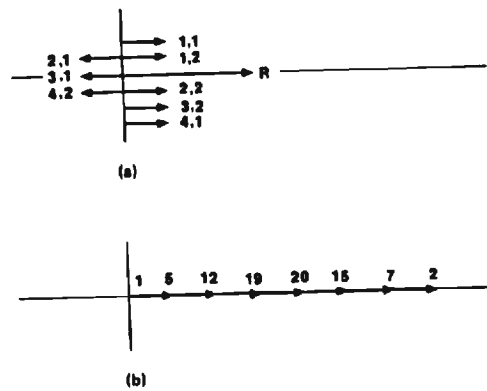


Fig.4

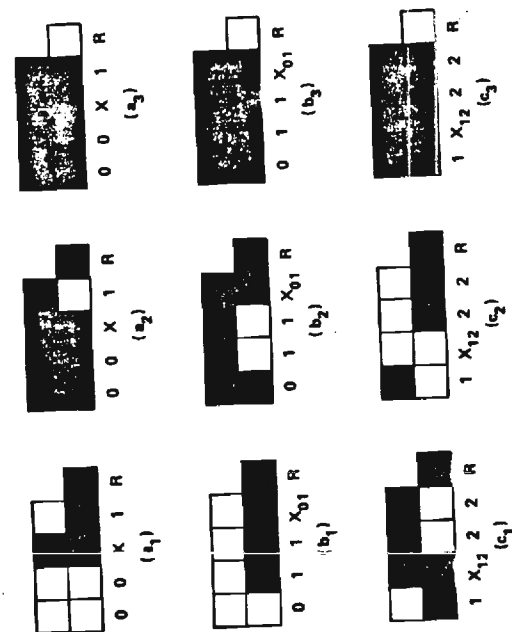


Fig.5

INTEGRATED OPTICAL GIVENS ROTATION DEVICE

M. M. Mirsalehi, T. K. Gaylord, and E. I. Verriest

School of Electrical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

(Received November , 1985)

The Givens rotation operation occupies a central role in linear algebraic signal processing. A lithium niobate integrated optical coherent implementation of an elementary rotation matrix device based on thick grating diffraction to perform this operation is proposed. It is shown that existing electro-optic phase shifting and grating diffraction devices can be combined to produce a very fast Givens rotation device.

1. Introduction

Very high speed data processing systems are needed in areas such as adaptive antenna beam forming, artificial intelligence, remote sensing, ultra-high resolution image processing, control of communication networks, air traffic control, synthetic aperture radar imaging, missile guidance, defense early warning systems, and simulation problems such as aerodynamic modeling and weather prediction requiring the solution of the Navier-Stokes equation. Real-time calculations for these highly complex and computationally intensive types of problems are largely beyond the capabilities of present day computing systems.

Monolithic integrated optical (guided wave) circuits offer the promise of very high speed processing. Devices that have been implemented in integrated optics include spectrum analyzers, analog-to-digital converters, convolvers, and correlators. In the present work, a lossless integrated optical implementation of an elementary rotation matrix ("Givens rotation" or "Jacobi rotation")^{1,2} device that operates on optical amplitude is proposed. This device uses electro-optic grating diffraction and electro-optic phase modulator devices to achieve the required multiplication, addition, and subtraction operations in the Givens rotation.

II. The Need for Givens Rotation

Solutions of the above listed problems can always be expressed in terms of linear algebra matrix-based algorithms.^{3,4} The types of operations that are needed include matrix-vector multiplication, matrix-matrix multiplication, matrix inversion, solution of linear equations, solution of least square problems, singular value decomposition, the discrete Fourier transform, and calculation of eigenvalues and eigenvectors. All of these calculations may be performed by using the Givens rotation operation, repeated many times over many elements. For example, a set of linear equations of arbitrary size can be solved using the Givens rotation to triangularize the matrix of coefficients, followed by backsubstitution to determine the unknowns.

The use of arrays of integrated optical devices and fiber optics have previously been proposed to implement systolic lattice filters.^{5,6} A form of the lattice (or ladder) filter structure, described by the square-root-normalized lattice equations, has a natural interpretation in terms of rotations. These structures are widely used for prediction and filtering in the areas of speech processing, channel equalization, seismic data interpretation, and electroencephalogram (EEG) analysis. The implementation of these filters involves a cascade of elementary sections, each consisting of a Givens rotation

and time delays.⁷ These algorithms are also related to the Schur-Cohn stability test.

III. Givens Rotation

The elementary rotation matrix may be expressed as

$$\begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}. \quad (1)$$

The flow graph corresponding to Eq. (1) is shown in Fig. 1. The Givens orthogonalization⁸ (or Givens rotation) operation is obtained when $\sin \psi$ and $\cos \psi$ are found such that $d = 0$. This operation can be used to make a particular element of a vector be zero. In fact, all but one entry of a vector can be made zero by successive Givens rotations (involving different entries of the vector). In matrix triangularization, $N-1$ rotations involving entries 1 and J ($J = 2, \dots, N$) are applied to "zero" the first column, i.e. to transform (rotate) the first column into the vector $[x, 0, \dots, 0]$. The same rotations (in the same order) are used to transform columns 2 to N . The triangularization of the $N \times N$ matrix is accomplished recursively by then zeroing the first column of the resulting lower right $N-1 \times N-1$ submatrix and so forth. Subsequent operations do not change the values in previously zeroed columns. This algorithm lends itself naturally to cascaded or pipelined hardware implementations. Due to the nonlinear $\sin \psi$ and $\cos \psi$ functions, the Givens rotation operation consumes a

significant amount of time and/or semiconductor material when implemented in digital electronics. This remains a problem, even though efficient bit-recursive methods using simple shift and add operations known as Coordinate Rotation Digital Computing (CORDIC) have been developed.⁷

IV. Device Configuration

The Givens rotation operation and lattice filtering simulate wave propagation phenomena. They can be modeled as lossless transmission line structures. Thus it is natural to consider wave propagation effects in constructing these devices. A coherent integrated optics implementation of an elementary rotation matrix device that operates on optical amplitude is proposed in this paper. This device uses electro-optic grating diffraction and phase shifting to achieve the required multiplications and summations in the Givens rotation operation.

The multiplications of the input amplitudes by sine and cosine are accomplished naturally and straightforwardly via diffraction by a "thick" transmission phase grating⁸ induced by a voltage applied to periodic metallic electrodes on the surface of the device. This multiplication of amplitudes by sine and cosine using voltage-induced grating diffraction is distinctly different from the multiplication of intensities by arbitrary numbers as used in integrated optical implementations of vector subtraction, vector scalar product, and matrix-vector product.⁹⁻¹¹ In the latter applications, a voltage that is proportional to the arcsine

of the square root of the multiplier must be precalculated. In contrast to this, the present device uses the natural sine and cosine multiplication characteristics of a thick phase grating directly.

The summations in the Givens rotation are achieved by coherently combining the output waves. The phases of the waves are adjusted with electro-optic phase shifters to achieve the required addition and subtraction indicated in Eq. (1). For binary numbers, the subtraction process is analogous to the EXCLUSIVE OR operation performed with thick holograms that has been previously analyzed¹² and experimentally demonstrated.¹³

The operating principle of the device is illustrated in Fig. 2. An optical wave of amplitude "a" is incident at the first Bragg angle upon a thick grating producing a transmitted amplitude of

$$S_0 a = a \cos \psi \quad (2)$$

and a diffracted amplitude of

$$S_1 a = a \exp(j\Gamma_g) \sin \psi \quad (3)$$

as shown in Fig. 2a. These amplitudes are referenced to the $x = 0$ origin at the output side of the grating. The transmitted amplitude at this point has been arbitrarily taken to be positive real. The phase factor $\exp(j\Gamma_g)$ that appears in the

diffracted amplitude, Eq. (3), represents the phase difference between these waves at the $x = 0$ output point. The sinusoidal grating refractive index may be expressed as

$$n(x) = n_0 + n_1 \cos(Kx + \phi_n) \quad (4)$$

where n_0 is the average index, n_1 is the amplitude of the index modulation, K is the magnitude of the grating vector ($K = 2\pi/\Lambda$), Λ is the grating period, and ϕ_n represents the phase of the cosinusoidal grating with respect to the $x = 0$ origin. If, for example, the $x = 0$ origin is chosen so that $\phi_n = 0$, then the grating has the commonly treated cosinusoidal form $n(x) = n_0 + n_1 \cos(Kx)$. For this case, $\zeta_a = -\pi/2$ and the diffracted amplitude is the well known result $S_1 = -j \sin\psi$. In general, the phase angle of the diffracted wave is

$$\zeta_a = \phi_n - \pi/2 \quad (5)$$

Similarly, a mutually coherent optical wave of amplitude "b" is incident at the other first Bragg angle producing a transmitted amplitude of

$$S_{0b} = b \cos\psi \quad (6)$$

and a diffracted amplitude of

$$S_{1b} = b \exp(j\zeta_b) \sin\psi \quad (7)$$

as shown in Fig. 2b. The phase factor $\exp(j\zeta_b)$ that appears in the diffracted amplitude, Eq. (6), again represents the phase difference between these two waves at the $x = 0$ output

point. Since the "b" wave has a component in the $-x$ direction (compared to a component in the $+x$ direction for the "a" wave), the phase angle, ζ_b associated with the diffracted wave is not the same as ζ_a . For the general case, it is

$$\zeta_b = -\phi_n - \pi/2 \quad (8)$$

Coherently combining these two diffraction processes and including two external phase shifts, Γ_1 and Γ_2 , produces the device shown in Fig. 2c. The output amplitudes are

$$c' = a \cos\psi + b \exp[j(\Gamma_1 + \zeta_b)] \sin\psi, \quad (9)$$

and

$$d' = a \exp[j(\zeta_a + \Gamma_2)] \sin\psi + b \exp[j(\Gamma_1 + \Gamma_2)] \cos\psi. \quad (10)$$

If $(\Gamma_1 + \zeta_b)$ is an integer multiple of 2π , if $(\zeta_a + \Gamma_2)$ is an odd integer multiple of π , and if $(\Gamma_1 + \Gamma_2)$ is an integer multiple of 2π , then $c' = c$ and $d' = d$, the values corresponding to those given by the elementary rotation matrix operation, Eq. (1). This may be accomplished by setting the external phase shifters so that

$$\Gamma_1 = \phi_n + \pi/2, \quad (11)$$

and

$$\Gamma_2 = -\phi_n - \pi/2, \quad (12)$$

and thus $\Gamma_2 = -\Gamma_1$.

The "a" and "b" waves combine to form an interference pattern at the location of the grating. In the absence of any external phase shifts and for "a" and "b" both representing positive numbers, the waves are in phase and a maximum in the interference pattern is taken to be the $x = 0$ origin. The interdigitated-electrode voltage-induced cosinusoidal grating may initially have an arbitrary phase, ϕ_n , with respect to this interference pattern. Equations (11) and (12) give the required external phase shifts in order to produce the outputs "c" and "d" given by the elementary rotation matrix, Eq.(1). Some values of ζ_a , ζ_b , Γ_1 , and Γ_2 are given in Table I for several representative values of ϕ_n . As can be seen from the table, if the grating formed by the interdigitated electrodes is a positive sine grating, then no external phase shifts are needed. This occurs if a bright fringe in the interference pattern occurs at a position in the grating where $n(x) = n_0$ and the index is increasing in the positive x direction. This configuration gives the required addition in the "c" and the required subtraction in the "d" output. Ideally, the interdigitated electrodes should be fabricated with this relationship to the interference pattern. However, for any position of the grating electrodes, the external phase shift can be adjusted so that the correct phase relationship is established to produce the elementary rotation operation.

A schematic top-view of an integrated optical implementation of this device is shown in Fig. 3. The input light signals of amplitudes "a" and "b" are guided as TM modes in channel waveguides. The output guided wave amplitudes are "c" and "d."

V. Device Elements

Channel Waveguides. The optical paths are conventional single-lateral-mode channel waveguides typically about 8 microns wide that are formed in the z-cut lithium niobate by titanium in-diffusion.

Coherent Inputs. The input signals are two mutually coherent, monochromatic TM guided waves. These would probably be derived from the same laser source. The amplitudes of these waves represent the numbers "a" and "b" which individually may be positive or negative. The quantities may be expressed as $a = |a| \exp(j\phi_a)$ and $b = |b| \exp(j\phi_b)$ where ϕ_a and ϕ_b are either 0 (for a positive number) or π (for a negative number). If "a" and "b" are of the same sign, the waves are in phase. If the numbers have opposite signs, the waves are 180 degrees out of phase.

The use of TM guided modes (polarization perpendicular to the surface of the device) as opposed to TE guided modes allows a lower grating modulation voltage (see below) to be used. This choice also avoids possible polarization rotation effects¹⁴⁻¹⁶ due to the bulk photovoltaic effect in

lithium niobate because the polarization is already parallel to the optic axis.

Coherently Coupled Bends. The device shown in Fig. 3 can be constructed with straight waveguides in a simple "X" configuration. However, if curved waveguides are used, as illustrated in Fig. 3, optical power losses associated with smooth circular bends¹⁷ may be avoided by using a series of straight waveguide segments of equal length for the curved portions of the channel waveguides. The bend loss oscillates as a function of segment length due to coupling between guided and unguided modes.¹⁸⁻¹⁹ However, light coupled out of a guided mode into an unguided mode at a bend can be entirely coupled back into the guided mode at the next bend if their phase difference is an odd multiple of 180 degrees at that next bend.²⁰ This can be accomplished by correctly selecting the length of the segments. Using coherently coupled bends, a loss of 0.08 dB per 1 degree bend has been experimentally measured for titanium in-diffused channel waveguides.^{20,21}

Grating. The grating is oriented so that the first Bragg condition² is satisfied for both input waves for the wavelength used. For an angle of incidence, θ , the required grating period, Λ , is

$$\Lambda = \lambda / 2n_e \sin \theta, \quad (13)$$

where λ is the freespace wavelength and n_e is the principal extraordinary refractive index. For $\lambda = 1.0$ micron, $n_e =$

2.158, and $\theta = 10$ degrees, the grating period would be about 1.33 microns. The Bragg regime parameter²² may be defined as $\rho = \lambda^2 / \Lambda^2 n_e n_i$ where n_i is the amplitude of the refractive index grating. If ρ is sufficiently large, the transmitted and diffracted amplitudes are proportional to sine and cosine as given by Eqs. (2), (3), (6), and (7). In grating diffraction, the parameter ψ is the grating strength parameter.²³ For TM guided modes, $\psi = \pi n_i d_g / \lambda \cos \theta$, where d_g is the grating thickness. The electric field component in the z direction (optic axis direction), E_z , produced by the interdigitated electrodes induces a refractive index grating whose amplitude is approximately $n_i = n_e^3 r_{33} E_z / 2$ where r_{33} is the element of the electro-optic tensor for z-polarized light and an applied electric field in the z direction. The magnitude of the applied electric field produced by the applied grating voltage, V_g , is approximately $2V_g / \Lambda$. Thus the transmitted and diffracted amplitudes produced by a positive sinusoidal grating may be expressed as

$$S_0 = s \cos(\pi d_g n_e^3 r_{33} V_g / \lambda \Lambda \cos \theta), \quad (14)$$

$$S_1 = -j s \sin(\pi d_g n_e^3 r_{33} V_g / \lambda \Lambda \cos \theta). \quad (15)$$

These amplitudes contain the desired cosine and sine multiplications. Integrated optical interdigitated-electrode electro-optic gratings for

intensity modulation and switching applications have been constructed and analyzed by numerous investigators.²³⁻²⁷

Phase Shifters. One arm of the device contains electro-optic phase shifters to adjust the phase to produce the desired real addition at the "c" output and the real subtraction at the "d" output. These devices also utilize E_x to change the extraordinary refractive index and thus change the optical path length.²⁸ For a phase shift of Γ_1 , the required electric field is

$$E_x = \lambda \Gamma_1 / \pi n_E^3 \Gamma_{33} L, \quad (16)$$

where L is the length of the electrodes. Since $\Gamma_2 = -\Gamma_1$, the phase shifter voltages will be equal in magnitude and opposite in sign.

Waveguide Crossing. Intersecting channel waveguides are capable of operating so that there is no net transfer of power from one waveguide to the other. In the present case, this is required for a grating voltage, V_g , of zero. With the addition of electro-optic modulators at the intersections, 2x2 switches have been constructed with one set of crossing channel waveguides^{24,29} and arrays of switches have been fabricated with multiple crossing waveguides.^{25,30} In the absence of external modulation, the coupling of power between intersecting waveguides oscillates as a function of the angle of intersection between the channel waveguides. The fraction of the optical amplitude

in a channel waveguide that remains in that waveguide is given approximately by³¹ $f = \cos[(1/2 \Lambda_c) \cot(\alpha/2)]$ where Λ_c is the coupling period and α is the intersection angle of the channel waveguides. The quantity Λ_c is a function of the waveguide material and geometry. Thus for zero crosstalk, the intersection angle is chosen so that $\alpha = 2 \cot^{-1}(2m\Lambda_c)$ where m is an integer. A high level of isolation between intersecting channel waveguides has been experimentally shown for TM modes.^{28,31}

VI. Discussion

The Givens rotation is a key operation in linear algebraic signal processing. The Givens rotation device described in this work is composed entirely of electro-optic waveguide devices. It could be constructed by 1) fabricating the channel waveguides, 2) growing a buffer layer over the surface, and 3) depositing the metal electrodes. Thus it has the potential for easy fabrication. The phase shifter voltages, $V_{e=1}$ and $V_{e=2}$, can be adjusted to give the desired performance. These voltages would then remain fixed. The grating voltage, V_g , would be initially varied until $d = 0$ was obtained. This voltage would then remain constant while new matrix element optical inputs ("a" and "b") are applied and the corresponding matrix element optical outputs ("c" and "d") are obtained. However, greatly increased processing throughput could be obtained by having parallel arrays of these devices with the same

intensity modulation and switching applications have been constructed and analyzed by numerous investigators.²³⁻²⁷

Phase Shifters. One arm of the device contains electro-optic phase shifters to adjust the phase to produce the desired real addition at the "c" output and the real subtraction at the "d" output. These devices also utilize E_x to change the extraordinary refractive index and thus change the optical path length.²⁸ For a phase shift of Γ_1 , the required electric field is

$$E_z = \lambda \Gamma_1 / \pi n_E^3 \Gamma_{33} L, \quad (16)$$

where L is the length of the electrodes. Since $\Gamma_2 = -\Gamma_1$, the phase shifter voltages will be equal in magnitude and opposite in sign.

Waveguide Crossing. Intersecting channel waveguides are capable of operating so that there is no net transfer of power from one waveguide to the other. In the present case, this is required for a grating voltage, V_g , of zero. With the addition of electro-optic modulators at the intersections, 2x2 switches have been constructed with one set of crossing channel waveguides^{24,29} and arrays of switches have been fabricated with multiple crossing waveguides.^{25,30} In the absence of external modulation, the coupling of power between intersecting waveguides oscillates as a function of the angle of intersection between the channel waveguides. The fraction of the optical amplitude

in a channel waveguide that remains in that waveguide is given approximately by³¹ $f = \cos[(1/2 \Lambda_c) \cot(\alpha/2)]$ where Λ_c is the coupling period and α is the intersection angle of the channel waveguides. The quantity Λ_c is a function of the waveguide material and geometry. Thus for zero crosstalk, the intersection angle is chosen so that $\alpha = 2 \cot^{-1}(2m\Lambda_c)$ where m is an integer. A high level of isolation between intersecting channel waveguides has been experimentally shown for TM modes.^{28,31}

VI. Discussion

The Givens rotation is a key operation in linear algebraic signal processing. The Givens rotation device described in this work is composed entirely of electro-optic waveguide devices. It could be constructed by 1) fabricating the channel waveguides, 2) growing a buffer layer over the surface, and 3) depositing the metal electrodes. Thus it has the potential for easy fabrication. The phase shifter voltages, $V_{e=1}$ and $V_{e=2}$, can be adjusted to give the desired performance. These voltages would then remain fixed. The grating voltage, V_g , would be initially varied until $d = 0$ was obtained. This voltage would then remain constant while new matrix element optical inputs ("a" and "b") are applied and the corresponding matrix element optical outputs ("c" and "d") are obtained. However, greatly increased processing throughput could be obtained by having parallel arrays of these devices with the same

grating voltage, V_g , applied to all of them. This would allow simultaneous parallel computation of all revised matrix elements associated with the zero element produced.

Indeed, the Givens rotation device could be used as a fundamental building block in lattice filters, wavefront processors,²² and a variety of other processing structures. In these devices the grating voltage would remain fixed and would not have to be varied. The device as schematically shown in Fig. 3 has codirectional data flow as is needed in a ladder implementation of finite impulse response (FIR) filtering. However, with minor reconfiguration as shown in Fig. 4, the device could be used with contradirectional data flow as is needed with a ladder implementation of infinite impulse response (IIR) filtering. The optical delay time associated with the device would be on the order of 25 to 50 picoseconds and thus the device could potentially function at high speeds. In lattice filtering applications, the required time delays between sections could be implemented by interconnecting the lithium niobate channel waveguides with single-mode fibers as has been done in communications applications.^{23,24} The final output numbers may be obtained by coherent detection of the wave amplitudes and phases using optical homodyne (local oscillator frequency is the same as the signal frequency) interference techniques.²⁵⁻²⁷

The devices that must be integrated to produce the proposed Givens rotation device have previously been experimentally constructed and reported in the literature.

Although lithium niobate technology has been referenced throughout this discussion, the same rotation matrix device structure could also be constructed in integrated optical form using an electro-optic semiconductor material such as gallium arsenide. Bulk optical implementations, though not as fast, would also be possible.

This work was supported in part by a grant from the Joint Services Electronics Program and by a grant from the Strategic Defense Initiative Office administered through the Office of Naval Research.

References

1. W. Givens, "Computation of plane unitary rotations transforming a general matrix to triangular form," SIAM J. Appl. Math 6, 26 (1958).
2. G. H. Golub and C. F. Van Loan, *Matrix Computations* (Johns Hopkins Univ. Press, Baltimore, 1983).
3. T. Kailath, "Signal processing in the VLSI era," in *VLSI and Modern Signal Processing* (S. Y. Kung, H. J. Whitehouse, and T. Kailath, eds., Prentice-Hall, Englewood Cliffs, NJ, 1985).
4. K. Bromley, "An interview with Keith Bromley on signal processing," *Optical Engineering Reports*, 1 (February 1985).
5. B. Moslehi, J. W. Goodman, M. Tur, and H. J. Shaw, "Fiber-optic lattice signal processing," *Proc. IEEE* 72, 909 (1984).
6. M. Nazarathy and J. W. Goodman, "Systolic signal processing with integrated optical coupled-wave device arrays," in *Optical Computing Technical Digest* (Opt. Soc. Am., Washington, DC, 1985).
7. J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. Electron. Comput.* 8, 330 (1959).
8. H. Kogelnik, "Coupled wave theory for thick hologram gratings," *Bell Syst. Tech. J.* 48, 2909 (1969).

9. C. M. Verber, R. P. Kenan, and J. R. Busch, "Design and performance of an integrated optical digital correlator," *J. Lightwave Tech.* LT-1, 256 (1983).
10. C. M. Verber, "Integrated-optical approaches to numerical optical processing," *Proc. IEEE* 72, 942 (1984).
11. C. M. Verber, "Integrated optical architectures for matrix multiplication," *Opt. Engr.* 24, 19 (1985).
12. C. C. Guest and T. K. Gaylord, "Truth-table look-up optical processing utilizing binary and residue arithmetic," *Appl. Opt.* 19, 1201 (1980).
13. C. C. Guest, M. M. Mirsalehi, and T. K. Gaylord, "EXCLUSIVE OR processing (binary image subtraction) using thick Fourier holograms," *Appl. Opt.* 23, 3444 (1984).
14. E. M. Zolotov, P. G. Kazanskii, and V. A. Chernykh, "Photoinduced polarization conversion in Ti:LiNbO_3 channel waveguides," *Sov. Tech. Phys. Lett.* 7, 397 (1981).
15. H. A. Haus, E. P. Ippen, A. Lattes, C. Gabriel, and F. J. Leonberger, "Double degenerate Four-Wave Mixing in LiNbO_3 Waveguides," *Appl. Phys. B* 28, 161 (1982).
16. J. F. Lam and H. W. Yen, "Dynamics of optical TE to TM mode conversion in LiNbO_3 channel waveguides," *Appl. Phys. Lett.* 45, 1172 (1984).
17. L. D. Hutcheson, I. A. White, and J. J. Burke, "Comparison of bending losses in integrated optical

circuits," Opt. Lett. 5, 276 (1980).

18. H. F. Taylor, "Power loss at directional change in dielectric waveguides," Appl. Opt. 13, 642 (1974).
19. H. F. Taylor, "Losses at corner bends in dielectric waveguides," Appl. Opt. 16, 711 (1977).
20. L. M. Johnson and F. J. Leonberger, "Low-loss LiNbO₃ waveguide bends with coherent coupling," Opt. Lett. 8, 111 (1983).
21. R. A. Becker and L. M. Johnson, "Low-loss multiple-branching circuit in Ti-indiffused LiNbO₃ channel waveguides," Opt. Lett. 9, 246 (1984).
22. T. K. Gaylord and M. G. Moharam, "Thin and thick gratings: Terminology clarification," Appl. Opt. 20, 3271 (1981).
23. C. M. Verber, V. E. Wood, R. P. Kenan, and N. F. Hartman, "Large-angle optical switching in waveguides in LiNbO₃," Ferroelect. 10, 253 (1976).
24. B. Chen and C. M. Meijer, "Bragg switch for optical channel waveguides," Appl. Phys. Lett. 33, 33 (1978).
25. R. A. Becker and W. S. C. Chang, "Electrooptical switching in thin film waveguides for a computer communication bus," Appl. Opt. 18, 3296 (1979).
26. E. M. Philipp-Rutz, R. Linares, and M. Fukuda, "Electrooptic Bragg diffraction switches in low cross-talk integrated-optics switching matrix," Appl. Opt. 21, 2189 (1982).
27. E. N. Glytsis, M. G. Moharam, and T. K. Gaylord,

"Refractive index distributions produced by interdigitated electrodes on electro-optic crystals," J. Opt. Soc. Am. A 2, xxx (1985).

28. R. C. Alferness, "Waveguide electrooptic modulators," IEEE Trans. Microwave Theo. Tech. MTT-30, 1121 (1982).
29. A. Neyer, "Electro-optic x-switch using single-mode Ti:LiNbO₃ channel waveguides," Electron. Lett. 19, 553 (1983).
30. L. McCaughan, "Long wavelength titanium-doped lithium niobate directional coupler optical switches and switch arrays," Opt. Engr. 24, 241 (1985).
31. E. E. Bergmann, L. McCaughan, and J. E. Watson, "Coupling of intersecting Ti:LiNbO₃ diffused waveguides," Appl. Opt. 23, 3000 (1984).
32. S. Y. Kung, "On supercomputing with systolic/wavefront array processors," Proc. IEEE 72, 867 (1984).
33. S. K. Korotky, G. Eisenstein, R. S. Alferness, J. J. Veselka, L. L. Buhl, G. T. Harvey, and P. H. Read, "Fully connectorized high-speed Ti:LiNbO₃ switch/modulator for time-division multiplexing and data encoding," J. Lightwave Technol. LT-3, 1 (1985).
34. S. K. Korotky, G. Eisenstein, A. H. Gnauck, B. L. Kasper, J. J. Veselka, R. C. Alferness, L. L. Buhl, C. A. Burrus, T. C. D. Huo, L. W. Stulz, K. C. Nelson, L. G. Cohen, R. W. Dawson, and J. C. Campbell, "4-Gb/s transmission experiment over 117 km of optical fiber

using a Ti:LiNbO_3 external modulator," J. Lightwave Technol. LT-3, 1027 (1985).

35. D. E. DeLonge and A. F. Dietrich, "Optical heterodyne experiments with enclosed transmission paths," Bell Syst. Tech. J. 47, 161 (1968).
36. T. Okoshi, "Recent progress in heterodyne/coherent optical fiber communications," J. Lightwave Technol. LT-2, 341 (1984).
37. D. W. Smith, "Coherent fiber optic communications," Laser Focus/Electro-Optics 21, 92 (November 1985).

Table I. Diffracted Wave Phase Factors (ζ_a and ζ_b). External Phase Shifts (Γ_1 and Γ_2) Required to Achieve Elementary Rotation Matrix Operation.*

$n(x)$	ϕ_n	ζ_a	ζ_b	Γ_1	Γ_2
$n_0 + n_1 \cos Kx$	0	$-\pi/2$	$-\pi/2$	$+\pi/2$	$-\pi/2$
$n_0 - n_1 \sin Kx$	$\pi/2$	0	π	π	π
$n_0 - n_1 \cos Kx$	π	$+\pi/2$	$+\pi/2$	$-\pi/2$	$+\pi/2$
$n_0 + n_1 \sin Kx$	$-\pi/2$	π	0	0	0

*The reference used throughout this paper, as shown in Fig. 2, is that the "a" wave has a component of its direction of propagation along the positive x axis.

FIGURE CAPTIONS

Fig. 1. Flow graph for the elementary rotation matrix operation.

Fig. 2. (a) Transmitted and diffracted amplitudes for "a" wave. (b) Transmitted and diffracted amplitudes for "b" wave. (c) Coherent combination of "a" and "b" diffraction with external phase shifters. For properly chosen phase shifts, c' and d' become the values given by the elementary rotation matrix. All transmitted and diffracted wave amplitudes are referenced to the $x = 0$ origin at the output side of the grating.

Fig. 3 Schematic of integrated optical elementary rotation matrix device (not to scale). The material is z-cut lithium niobate. The optical input amplitudes are "a" and "b"; the output amplitudes are "c" and "d." The grating voltage (V_g), and phase-shifter voltages ($V_{\pi/4}$ and $V_{\pi/2}$) are shown.

Fig. 4 (a) Codirectional form of rotation matrix device for FIR lattice filtering. (b) Contradirectional form of rotation matrix device for IIR filtering.

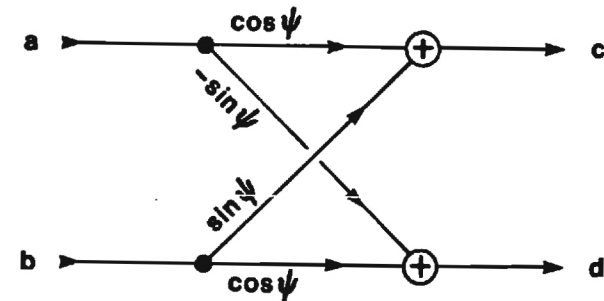


Fig. 1

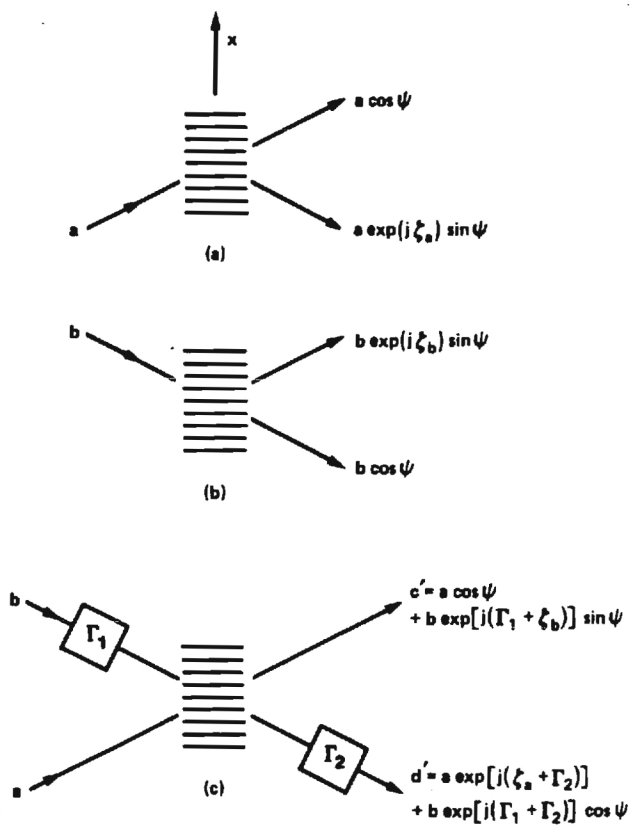


Fig. 2

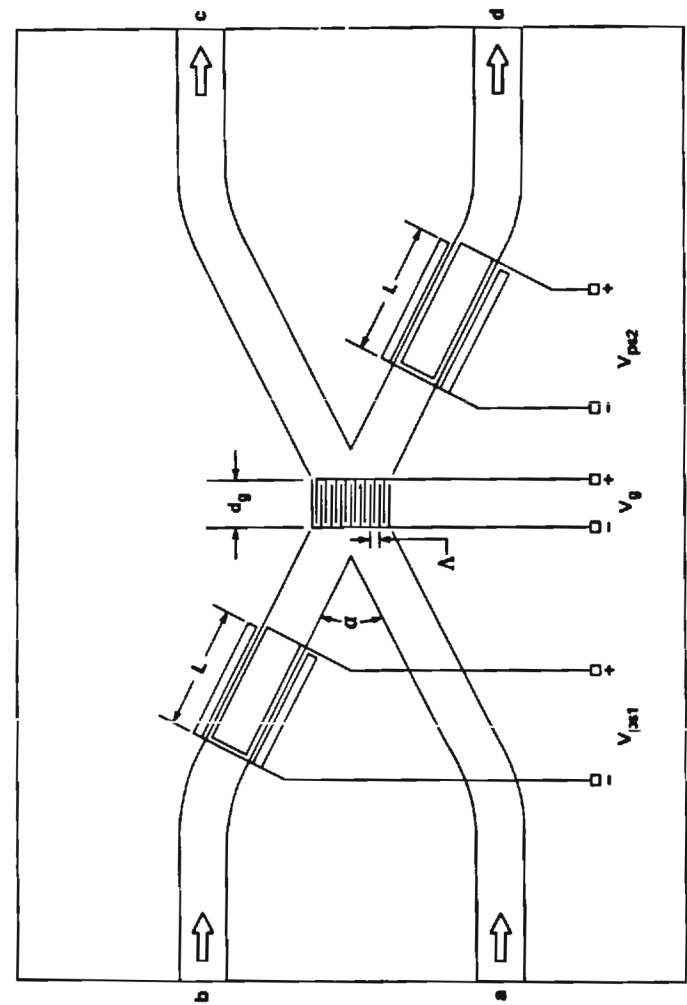
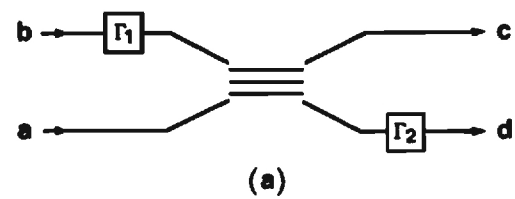
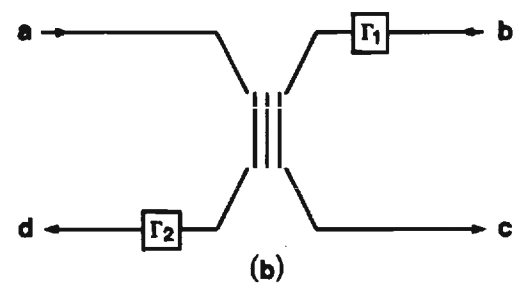


Fig. 3



(a)



(b)

Fig. 4

grating voltage, V_g , applied to all of them. This would allow simultaneous parallel computation of all revised matrix elements associated with the zero element produced.

Indeed, the Givens rotation device could be used as a fundamental building block in lattice filters, wavefront processors,²² and a variety of other processing structures. In these devices the grating voltage would remain fixed and would not have to be varied. The device as schematically shown in Fig. 3 has codirectional data flow as is needed in a ladder implementation of finite impulse response (FIR) filtering. However, with minor reconfiguration as shown in Fig. 4, the device could be used with contradirectional data flow as is needed with a ladder implementation of infinite impulse response (IIR) filtering. The optical delay time associated with the device would be on the order of 25 to 50 picoseconds and thus the device could potentially function at high speeds. In lattice filtering applications, the required time delays between sections could be implemented by interconnecting the lithium niobate channel waveguides with single-mode fibers as has been done in communications applications.^{23,24} The final output numbers may be obtained by coherent detection of the wave amplitudes and phases using optical homodyne (local oscillator frequency is the same as the signal frequency) interference techniques.²⁵⁻²⁷

The devices that must be integrated to produce the proposed Givens rotation device have previously been experimentally constructed and reported in the literature.

Although lithium niobate technology has been referenced throughout this discussion, the same rotation matrix device structure could also be constructed in integrated optical form using an electro-optic semiconductor material such as gallium arsenide. Bulk optical implementations, though not as fast, would also be possible.

This work was supported in part by a grant from the Joint Services Electronics Program and by a grant from the Strategic Defense Initiative Office administered through the Office of Naval Research.

References

1. W. Givens, "Computation of plane unitary rotations transforming a general matrix to triangular form," SIAM J. Appl. Math 6, 26 (1958).
2. G. H. Golub and C. F. Van Loan, *Matrix Computations* (Johns Hopkins Univ. Press, Baltimore, 1983).
3. T. Kailath, "Signal processing in the VLSI era," in *VLSI and Modern Signal Processing* (S. Y. Kung, H. J. Whitehouse, and T. Kailath, eds., Prentice-Hall, Englewood Cliffs, NJ, 1985).
4. K. Bromley, "An interview with Keith Bromley on signal processing," *Optical Engineering Reports*, 1 (February 1985).
5. B. Moslehi, J. W. Goodman, M. Tur, and H. J. Shaw, "Fiber-optic lattice signal processing," *Proc. IEEE* 72, 909 (1984).
6. M. Nazarathy and J. W. Goodman, "Systolic signal processing with integrated optical coupled-wave device arrays," in *Optical Computing Technical Digest* (Opt. Soc. Am., Washington, DC, 1985).
7. J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. Electron. Comput.* 8, 330 (1959).
8. H. Kogelnik, "Coupled wave theory for thick hologram gratings," *Bell Syst. Tech. J.* 48, 2909 (1969).

9. C. M. Verber, R. P. Kenan, and J. R. Busch, "Design and performance of an integrated optical digital correlator," *J. Lightwave Tech.* LT-1, 256 (1983).
10. C. M. Verber, "Integrated-optical approaches to numerical optical processing," *Proc. IEEE* 72, 942 (1984).
11. C. M. Verber, "Integrated optical architectures for matrix multiplication," *Opt. Engr.* 24, 19 (1985).
12. C. C. Guest and T. K. Gaylord, "Truth-table look-up optical processing utilizing binary and residue arithmetic," *Appl. Opt.* 19, 1201 (1980).
13. C. C. Guest, M. M. Mirsalehi, and T. K. Gaylord, "EXCLUSIVE OR processing (binary image subtraction) using thick Fourier holograms," *Appl. Opt.* 23, 3444 (1984).
14. E. M. Zolotov, P. G. Kazanskii, and V. A. Chernykh, "Photoinduced polarization conversion in Ti:LiNbO_3 channel waveguides," *Sov. Tech. Phys. Lett.* 7, 397 (1981).
15. H. A. Haus, E. P. Ippen, A. Lattes, C. Gabriel, and F. J. Leonberger, "Double degenerate Four-Wave Mixing in LiNbO_3 Waveguides," *Appl. Phys. B* 28, 161 (1982).
16. J. F. Lam and H. W. Yen, "Dynamics of optical TE to TM mode conversion in LiNbO_3 channel waveguides," *Appl. Phys. Lett.* 45, 1172 (1984).
17. L. D. Hutcheson, I. A. White, and J. J. Burke, "Comparison of bending losses in integrated optical

circuits," Opt. Lett. 5, 276 (1980).

18. H. F. Taylor, "Power loss at directional change in dielectric waveguides," Appl. Opt. 13, 642 (1974).
19. H. F. Taylor, "Losses at corner bends in dielectric waveguides," Appl. Opt. 16, 711 (1977).
20. L. M. Johnson and F. J. Leonberger, "Low-loss LiNbO₃ waveguide bends with coherent coupling," Opt. Lett. 8, 111 (1983).
21. R. A. Becker and L. M. Johnson, "Low-loss multiple-branching circuit in Ti-indiffused LiNbO₃ channel waveguides," Opt. Lett. 9, 246 (1984).
22. T. K. Gaylord and M. G. Moharam, "Thin and thick gratings: Terminology clarification," Appl. Opt. 20, 3271 (1981).
23. C. M. Verber, V. E. Wood, R. P. Kenan, and N. F. Hartman, "Large-angle optical switching in waveguides in LiNbO₃," Ferroelect. 10, 253 (1976).
24. B. Chen and C. M. Meijer, "Bragg switch for optical channel waveguides," Appl. Phys. Lett. 33, 33 (1978).
25. R. A. Becker and W. S. C. Chang, "Electrooptical switching in thin film waveguides for a computer communication bus," Appl. Opt. 18, 3296 (1979).
26. E. M. Philipp-Rutz, R. Linares, and M. Fukuda, "Electrooptic Bragg diffraction switches in low cross-talk integrated-optics switching matrix," Appl. Opt. 21, 2189 (1982).
27. E. N. Glytsis, M. G. Moharam, and T. K. Gaylord,

"Refractive index distributions produced by interdigitated electrodes on electro-optic crystals," J. Opt. Soc. Am. A 2, xxx (1985).

28. R. C. Alferness, "Waveguide electrooptic modulators," IEEE Trans. Microwave Theo. Tech. MTT-30, 1121 (1982).
29. A. Neyer, "Electro-optic x-switch using single-mode Ti:LiNbO₃ channel waveguides," Electron. Lett. 19, 553 (1983).
30. L. McCaughan, "Long wavelength titanium-doped lithium niobate directional coupler optical switches and switch arrays," Opt. Engr. 24, 241 (1985).
31. E. E. Bergmann, L. McCaughan, and J. E. Watson, "Coupling of intersecting Ti:LiNbO₃ diffused waveguides," Appl. Opt. 23, 3000 (1984).
32. S. Y. Kung, "On supercomputing with systolic/wavefront array processors," Proc. IEEE 72, 867 (1984).
33. S. K. Korotky, G. Eisenstein, R. S. Alferness, J. J. Veselka, L. L. Buhl, G. T. Harvey, and P. H. Read, "Fully connectorized high-speed Ti:LiNbO₃ switch/modulator for time-division multiplexing and data encoding," J. Lightwave Technol. LT-3, 1 (1985).
34. S. K. Korotky, G. Eisenstein, A. H. Gnauck, B. L. Kasper, J. J. Veselka, R. C. Alferness, L. L. Buhl, C. A. Burrus, T. C. D. Huo, L. W. Stulz, K. C. Nelson, L. G. Cohen, R. W. Dawson, and J. C. Campbell, "4-Gb/s transmission experiment over 117 km of optical fiber

using a Ti:LiNbO_3 external modulator," J. Lightwave Technol. LT-3, 1027 (1985).

35. D. E. DeLange and A. F. Dietrich, "Optical heterodyne experiments with enclosed transmission paths," Bell Syst. Tech. J. 47, 161 (1968).
36. T. Okoshi, "Recent progress in heterodyne/coherent optical fiber communications," J. Lightwave Technol. LT-2, 341 (1984).
37. D. W. Smith, "Coherent fiber optic communications," Laser Focus/Electro-Optics 21, 92 (November 1985).

Table I. Diffracted Wave Phase Factors (ζ_a and ζ_b). External Phase Shifts (Γ_1 and Γ_2) Required to Achieve Elementary Rotation Matrix Operation.*

$n(x)$	ϕ_n	ζ_a	ζ_b	Γ_1	Γ_2
$n_0 + n_1 \cos Kx$	0	$-\pi/2$	$-\pi/2$	$+\pi/2$	$-\pi/2$
$n_0 - n_1 \sin Kx$	$\pi/2$	0	π	π	π
$n_0 - n_1 \cos Kx$	π	$+\pi/2$	$+\pi/2$	$-\pi/2$	$+\pi/2$
$n_0 + n_1 \sin Kx$	$-\pi/2$	π	0	0	0

*The reference used throughout this paper, as shown in Fig. 2, is that the "a" wave has a component of its direction of propagation along the positive x axis.

FIGURE CAPTIONS

Fig. 1. Flow graph for the elementary rotation matrix operation.

Fig. 2. (a) Transmitted and diffracted amplitudes for "a" wave. (b) Transmitted and diffracted amplitudes for "b" wave. (c) Coherent combination of "a" and "b" diffraction with external phase shifters. For properly chosen phase shifts, c' and d' become the values given by the elementary rotation matrix. All transmitted and diffracted wave amplitudes are referenced to the $x = 0$ origin at the output side of the grating.

Fig. 3 Schematic of integrated optical elementary rotation matrix device (not to scale). The material is z-cut lithium niobate. The optical input amplitudes are "a" and "b"; the output amplitudes are "c" and "d." The grating voltage (V_g), and phase-shifter voltages ($V_{\pi=1}$ and $V_{\pi=2}$) are shown.

Fig. 4 (a) Codirectional form of rotation matrix device for FIR lattice filtering. (b) Contradirectional form of rotation matrix device for IIR filtering.

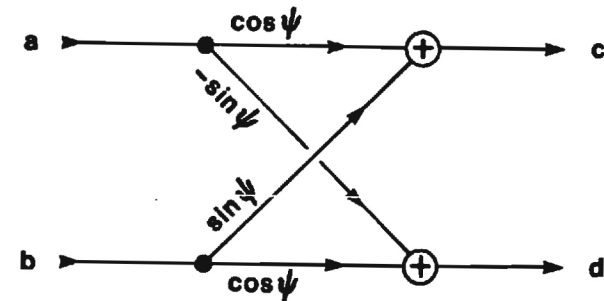


Fig. 1

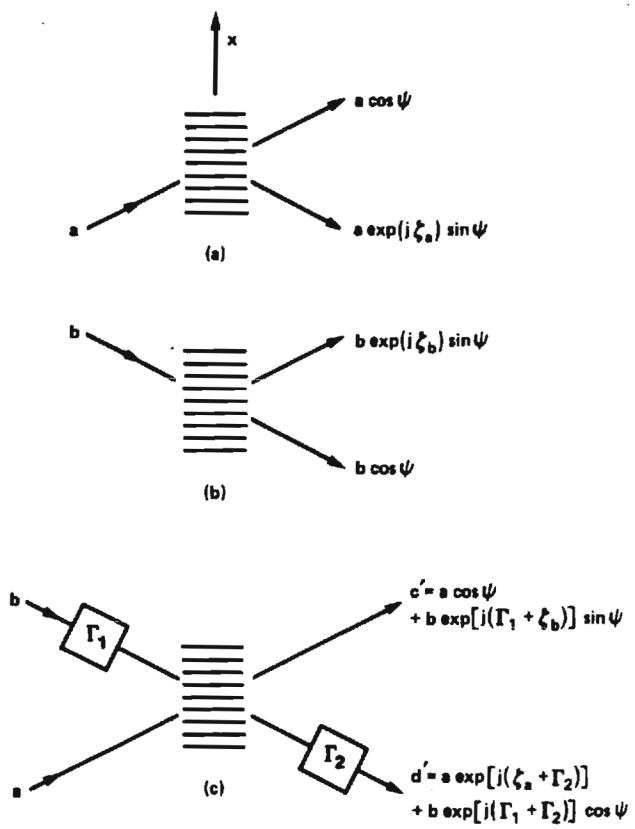


Fig. 2

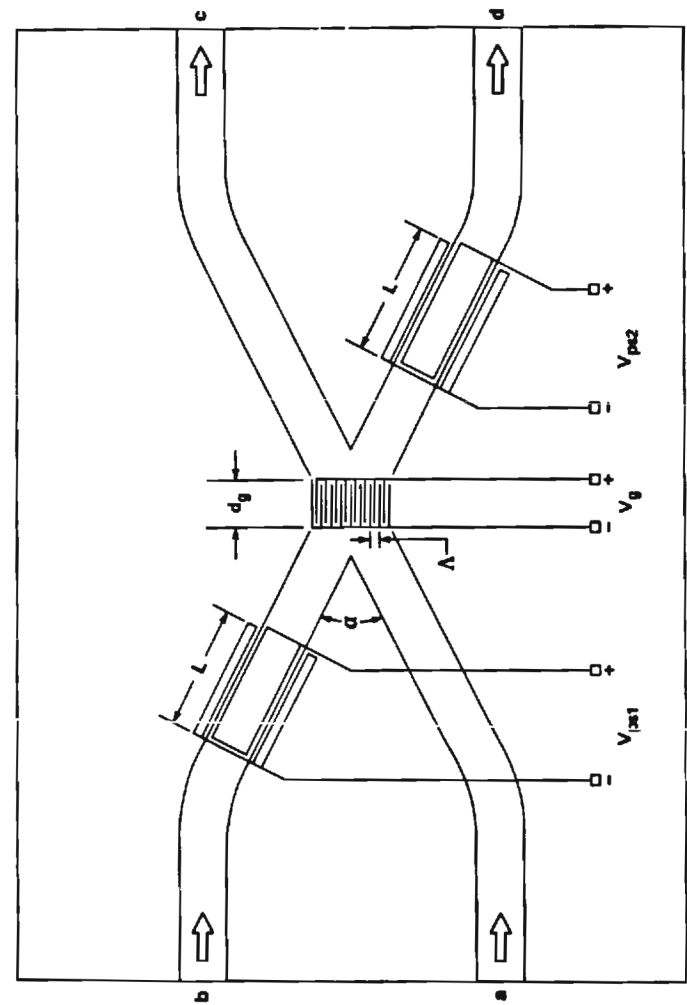
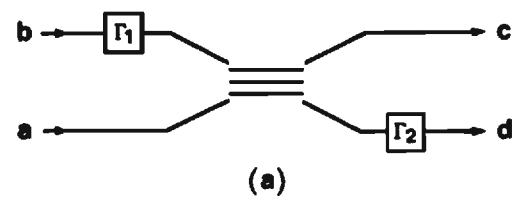
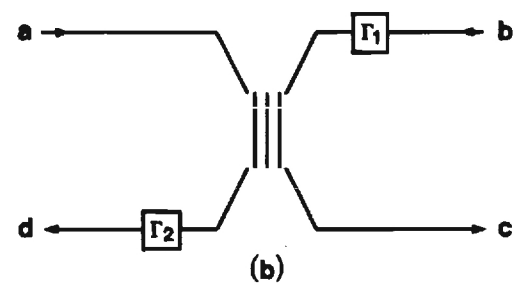


Fig. 3



(a)



(b)

Fig. 4